The Patent Office

U.S.N. 09/373980

INVESTOR IN PEOPLE

The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.

CERTIFIED COPY OF PRIORITY DOCUMENT

Signed [signature]

Dated 1 December 2003

Patents Form 1/77
Patents Act 1977
(Rule 16)

THE PATENT OFFICE
C
27 MAY 1997
RECEIVED BY HAND

The Patent Office

29MAY97 E277583-1 D02917
P01/7700 25.00 - 9710908.6

Request for grant of a patent

The Patent Office
Cardiff Road
Newport
Gwent NP9 1RH

1. Your reference
5219604/JJG

2. Patent Application Number
27 MAY 1997 9710908-6

3. Full name, address and postcode of the or of each applicant *(underline all surnames)*

Communication & Control Electronics Limited
Stirling House
Stirling Road
The Surrey Research Park
Guildford, Surrey GU2 5RF

Patents ADP number *(if known)* 6791636002

If the applicant is a corporate body, give the country/state of its incorporation
Country: ENGLAND
State:

4. Title of the invention
LOCAL COMMUNICATION SYSTEM

5. Name of agent
~~**Beresford & Co**~~ Fitzpatricks

"Address for Service" in the United Kingdom to which all correspondence should be sent
~~**2/5 Warwick Court**~~
~~**High Holborn**~~
~~**London WC1R 5DJ**~~
4 West Regent Street
Glasgow
G2 1RS

5/7/7
27/3/98

Patents ADP number ~~1826001~~ (95002)

6. Priority details

Country	Priority application number	Date of filing
GB	9622922.4	4 NOVEMBER 1996
GB	9703216.3	17 FEBRUARY 1997
GB	9704901.9	10 MARCH 1997

Patents Form 1/77

7. If this application is divided or otherwise derived from an earlier UK application give details

Number of earlier of application   Date of filing

8. Is a statement of inventorship and or right to grant of a patent required in support of this request?

YES

9. Enter the number of sheets for any of the following items you are filing with this form.

0	Continuation sheets of this form
62	Description
4	Claim(s)
0	Abstract
0	Drawing(s)

10. If you are also filing any of the following, state how many against each item.

0	Priority documents
0	Translations of priority documents
0	Statement of inventorship and right to grant of a patent *(Patents form 7/77)*
1	Request for preliminary examination and search *(Patents Form 9/77)*
0	Request for Substantive Examination *(Patents Form 10/77)*

11. I/We request the grant of a patent on the basis of this application

Signature Beresford & Co   Date 27 May 1997

BERESFORD & Co

12. Name and daytime telephone number of person to contact in the United Kingdom

JOHN JAMES GRAY

Tel: 0171-831-2290

DUPLICATE

LOCAL COMMUNICATION SYSTEM

A local communication system which combines source data (CD audio, MPEG video, telephone audio etc) with control commands in a low cost fibre network is available in the form of D2B Optical. For details, see for example the "Conan Technology Brochure" and the "Conan IC Data Sheet" available from Communication & Control Electronics Limited, Stirling House, Stirling Road, The Surrey Research Park, Guildford, Surrey GU2 5RF. An extract from the CONAN IC Data Sheet is attached hereto as an Appendix. See also German patent applications of Becker GmbH with filing numbers 19503206.3 (95P03), 19503207.1 (95P04), 19503209.8 (95P05), 19503210.1 (95P06), 19503212.8 (95P07), 19503213.6 (95P08), 19503214.4 (95P09) and 19503215.2 (95P10). "Conan" is a registered trade mark of Communication & Control Electronics Limited.

The present invention aims to enable expansion of the capacity of such a network, for use in vehicles and the like, towards a capacity necessary for higher bit-rate multimedia applications such as MPEG2 audio, MPEG2 video, Digital Audio Broadcasting (DAB), Digital Versatile Disk (DVD) and other data.

One proposed embodiment employs asynchronous transfer mode data transport for the various data types, accommodating both high bit rate and low bit rate data channels. However, the ATM packet is not limited to the conventional ATM packet of 5 bytes of header and 48 bytes of payload, but is adapted to provide more efficient use of bandwidth in the target applications.

Also, compared with D2B Optical (CONAN) technology,

it is proposed to use a more compact encoding technique for the fibre interface, such as the 4B5B or 8B10B encoding, currently used in FDDI (Fibre Distributed Data Interface) for metropolitan area networks (MANs).

Embodiments are proposed, in which each network frame includes subframes of at least two different modes: "mode 0" subframes provide compatibility with the fixed bit-rate (e.g. circuit-switched) CONAN technology, while simultaneously the "mode 1" subframe provides a group of bytes which can be allocated more freely to implement a variable bit-rate (e.g. packet-switched) channel. In other embodiments, a common source data channel is allocated dynamically between circuit-switched and packet-switched data.

These proposals and surrounding considerations are described in the following pages, together with some alternatives also proposed herein.

1. Introduction

This document comprises an outline proposal for a high speed D2B which works with an optical interface (Plastic Optical Fibre). The proposed device is known as the high speed D2B and is designated as C&C Electronics part number HSCI8001.

2. Objectives

The objectives of the HSCI8001 development are as follows

- To integrate many of the digital audio and video systems over a common data bus and come up with a suitable protocol that will be able to integrate them.
- Baseline for the High Speed Conan
  - Use of Plastic Optical Fibre
  - Low cost LED Transceiver technology
  - Bandwidth requirements > 25 Mbs
  - Ring Topology for ease of installation, low cost Fibre Optics components and low power consumption...

3.0 Requirement Constraints

3.1 Topology

- Point-to-point links
- Linear T bus
- Star system
- Reflective star
- Transmissive Star

3.2 Line encoding

There is a need for a more efficient encoding/decoding technique than what is currently used on the current Conan chip (BI-PHASE). One type

of encoding/ decoding technique that is currently used in FDDI which would be considered is the use of 4B5B encoding technique. This would provide the High speed network a reliable and efficient technique.

- BI Phase encoding for current Conan design
- 4B5B/8B10B

3.1 Data rate Requirements

- Parameters identified for the various digital audio video systems for integration over a common databus. The tabulated values indicate the some of the overriding parameters between systems..

Data Type	Variable Bandwidth Requirements	Common Frequency	Frame Structure
MPEG2 Audio	up to 912kbps	48kbps	
MPEG2 Video	1.55 to 4Mbps		
DAB	1.5Mbps?	48kHz	16-32 Bits
DVD	11.08Mbps		2048bytes Packet
CD	1.44Mbps	44.1kHz	16-24Bits
ATM (Use this as a Transport for above Data Types)			Similar to 5bytes of Header and 48 Bytes of Payload

3.4 Topology

The communications network forms a backbone of most architectural designs and hence the interconnections can influence the final design. Network topology fall mainly into three categories.

For the application that we are considering a Ring topology is still the preferred option as it means that the components available for the Optical network will still be relatively cheap and low power, rather then going to a relatively expensive star network.

These point are discussed further in the sections below. But it is highly likely that if in a large network environment if the system could not handle the delay

then a star network could be the way forward if the components become available for the Plastic Fibre optics and are relatively cheap.

Point-to-point links:

Point to point dedicated links are used in two ways. Firstly by a dedicated link between each communicating function. Secondly by a dedicate link between terminals and using electronics in each terminal to strip off data which is addressed to that particular terminal and pass any further data to other terminal.

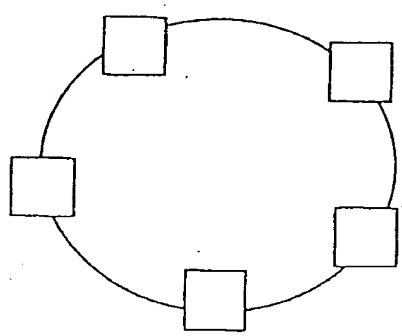


Fig. 1

Linear T- Bus

These are relatively simple to implement if using voltage mode or current mode coupling and provide a broadcast medium that allows all terminals to simultaneously listen to the traffic on the network. This has been used in many of the electrical broadcast bus designs, but is far less popular for optical systems as the excess loss at each junction leads to extremely large link power budgets.

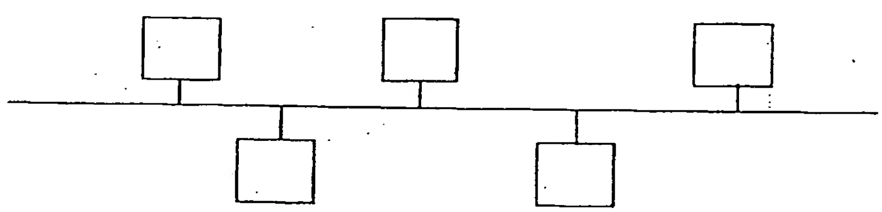


Fig.2

Stars

These can split light from a fibre into a number of other fibres in a reversible manner, that is 1:n or n:1. Many networks using fibre optics are implemented using a topology based on reflective and transmissive star couplers. But teh cost penalty is higher.

Reflective Star

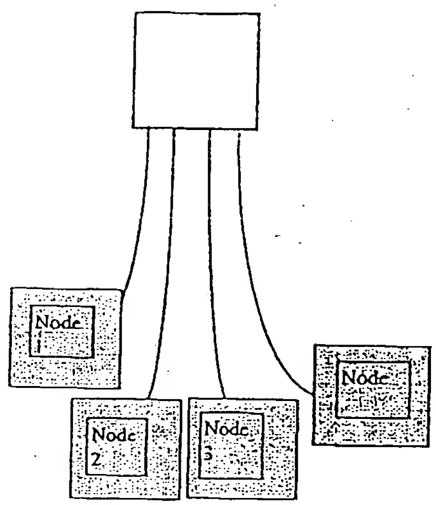


Fig. 3

Transmissive Star

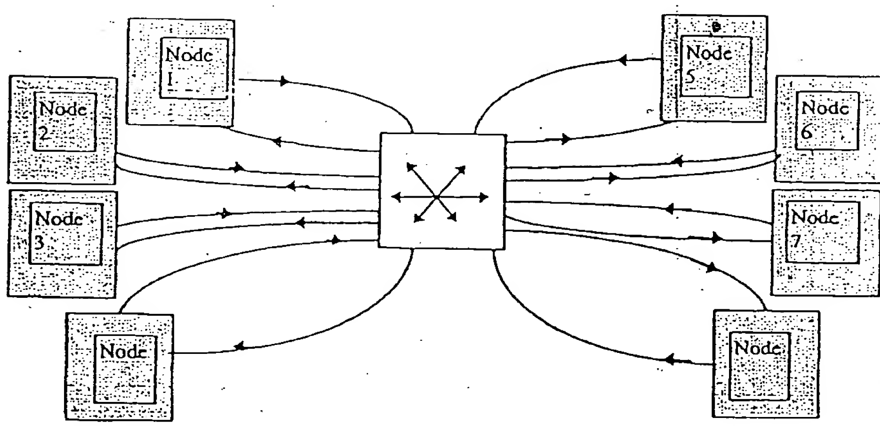


Fig. 4

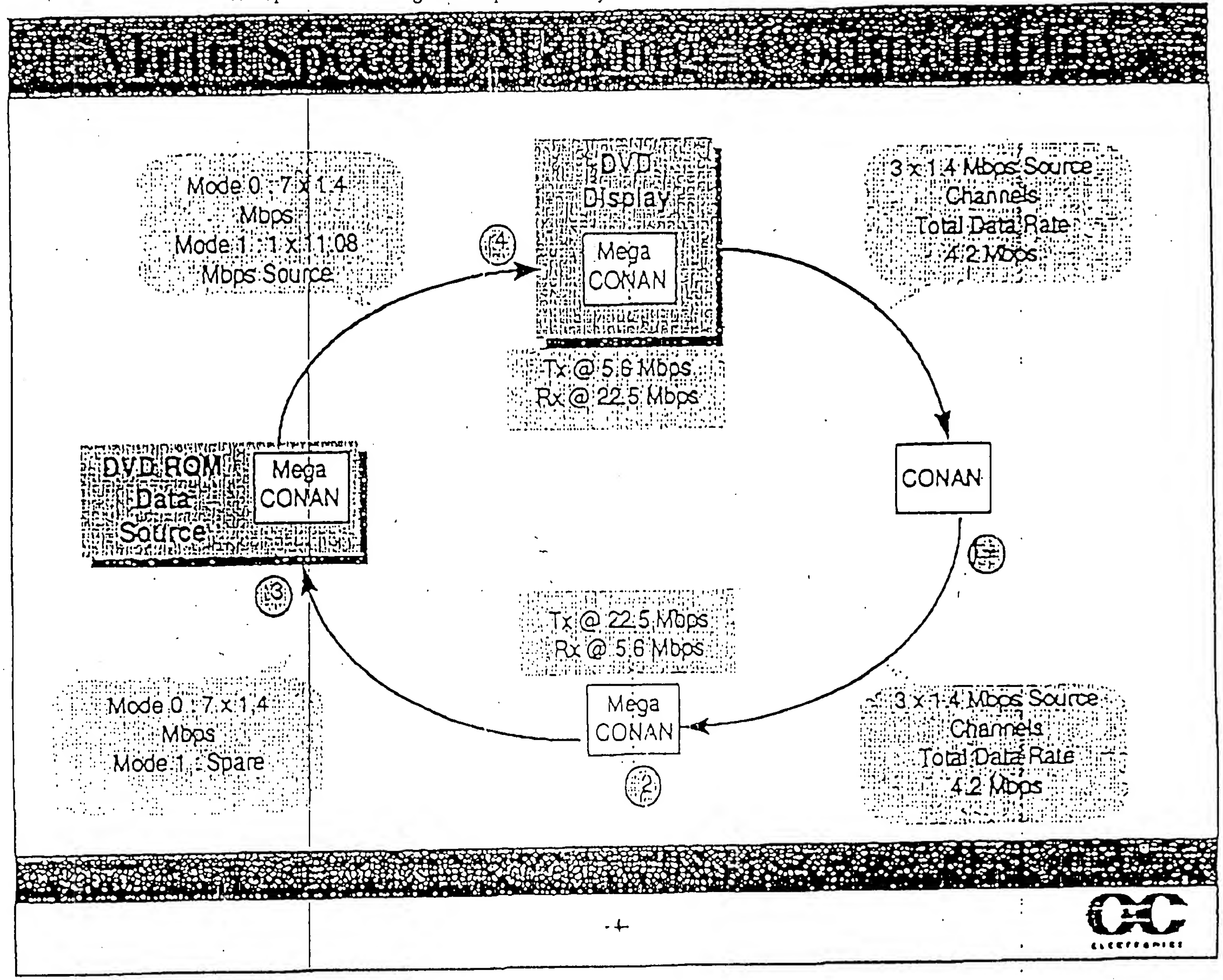
Detailed Proposal I

The following pages present a first embodiment of a High Speed D2B network offering high speed variable bit rate channels ("Mode 1" data) simultaneously with fixed rate channels ("Mode 0") compatible with the known "CONAN" technology (and with the proposed "SuperCONAN" technology having an increased capacity of fixed data rate channels).

A "MegaCONAN" chip is described, a key feature of which is the provision of dual processors. The first processor is a RISC processor corresponding to that used in the existing CONAN chip for implementing the D2B system with control of the fixed data rate ("Mode 0") data routing. The second processor, handing the asynchronous ("Mode 1") data is a digital signal processor (DSP) which can implement, for example sample rate conversion, audio DSP functions (such as Dolby AC3), and can interface to on-chip or off-chip for DVD and other data formats.

- Synchronous, connection oriented data transfer (mode 0).
- Asynchronous, connection-less data transfer (mode 1).
- Total bus capacity can be shared between modes 0 and 1.
- Bus running at an instant mixed frame rate in the range 16 - 50 kHz. Specific frame rates which are currently foreseen are CD at 44.1 kHz, DAB/DVD Audio at 48 kHz and telephony audio at 16 kHz.
- The total minimum source data rate is 4 times higher than for current D2B Optical
- For data transfer mode 1, the prioritisation of message and/or devices should depend on the application.
- Backwards compatibility with current D2B Optical.

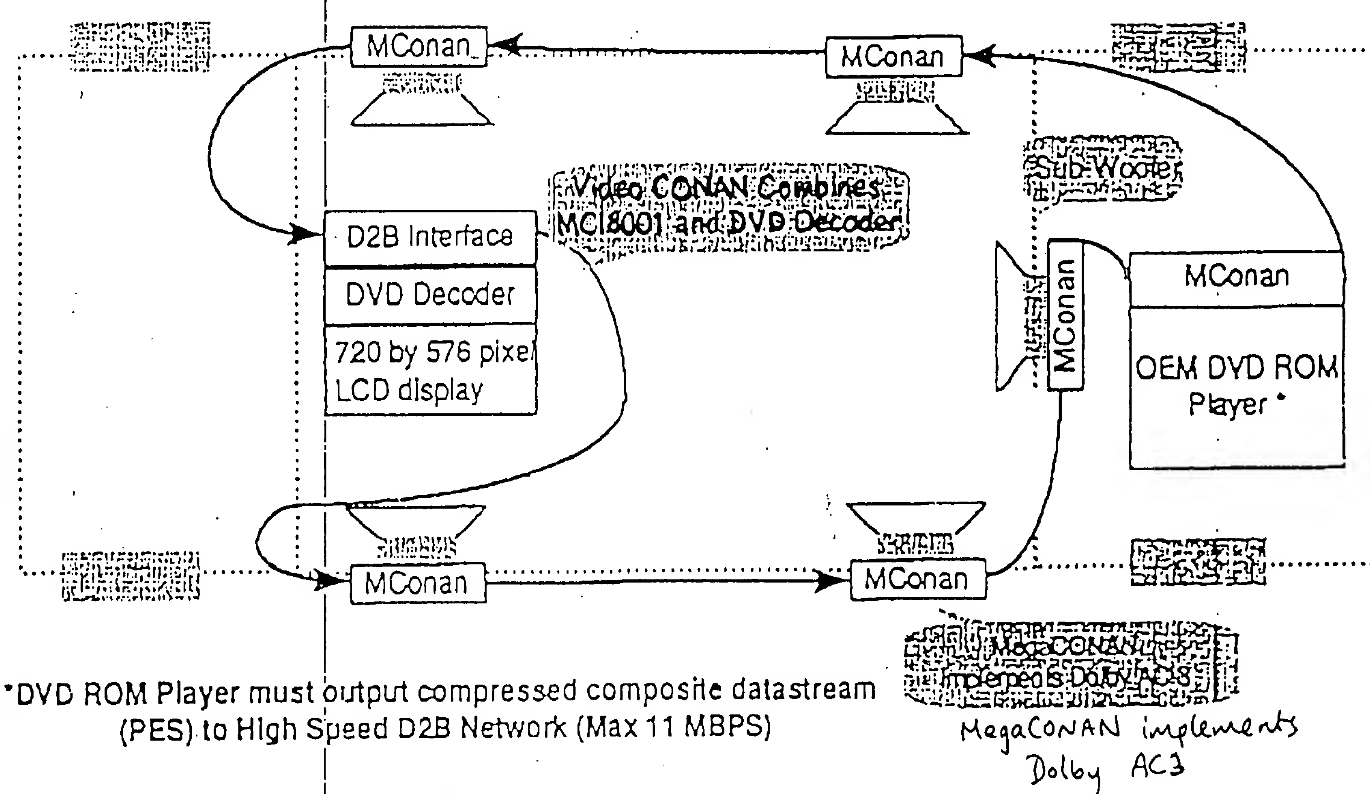
Multi-Speed D2B Ring - Compatibility



DVD requirements for High Speed D2B

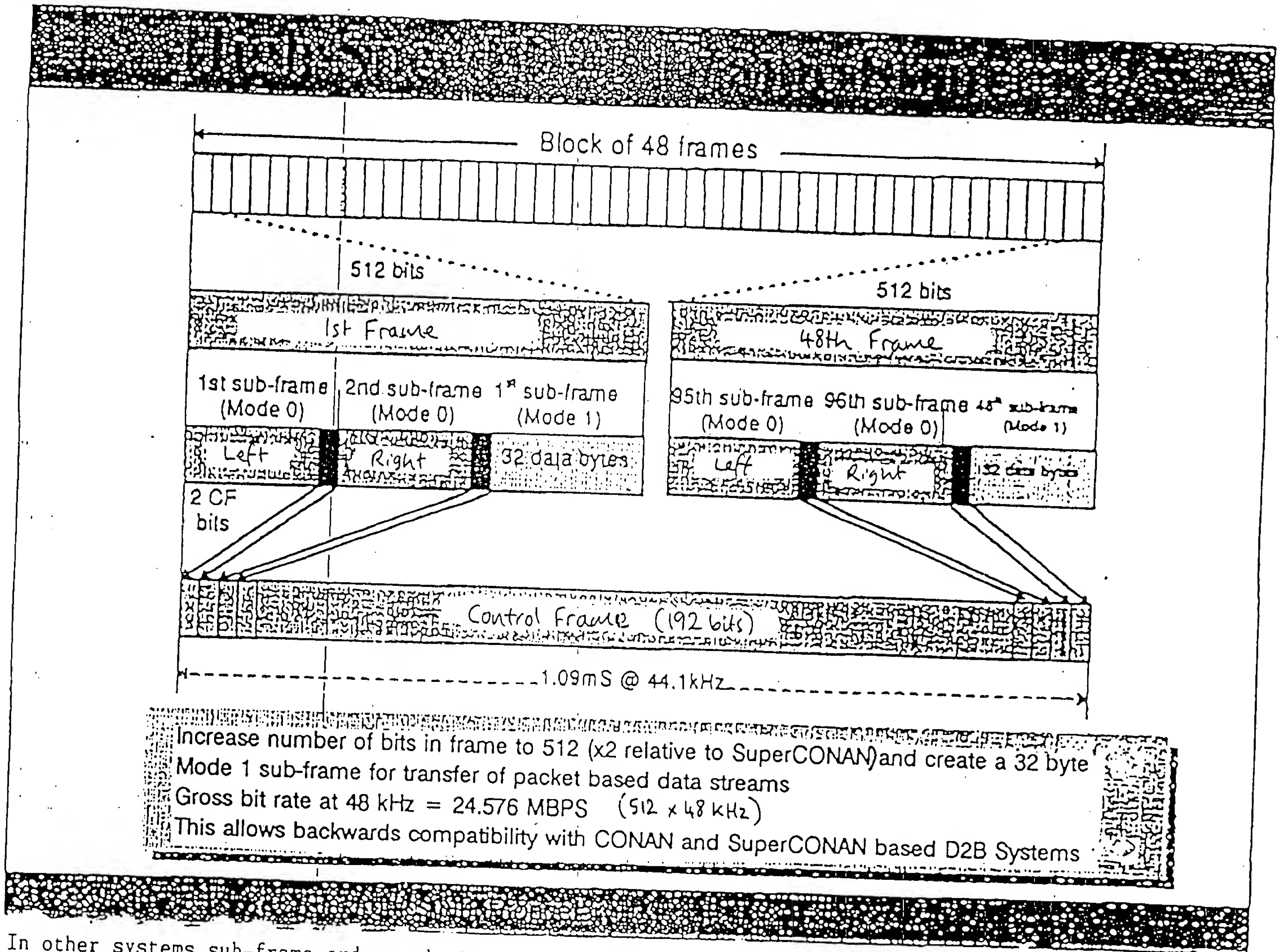
- DVD Summary
  - Maxlmum 17 GByte storage (c.f. 680MByte for CD)
  - Interactive video requires control info exchange
  - Typical image resolution corresponding to MPEG2 Main Level Main Profile : 720 (H) by 480 (V) pixels at 30 fps : 9.8 Mbps max
  - Dolby AC3 5.1 audio : 48 kHz sample rate : 448 kbps max
  - MPEG2 7.1 audio : 48 kHz sample rate : 912 kbps max
  - Average data rate of 4.69 MBPS (depending on content)
  - Peak information rate of 10.08 MBPS
  - Peak System layer rate of 11.08 MBPS
- DVD ROM Part 3 Specifies mapping of audio datastreams to IEC958

Possible Automotive DVD/D2B Topology



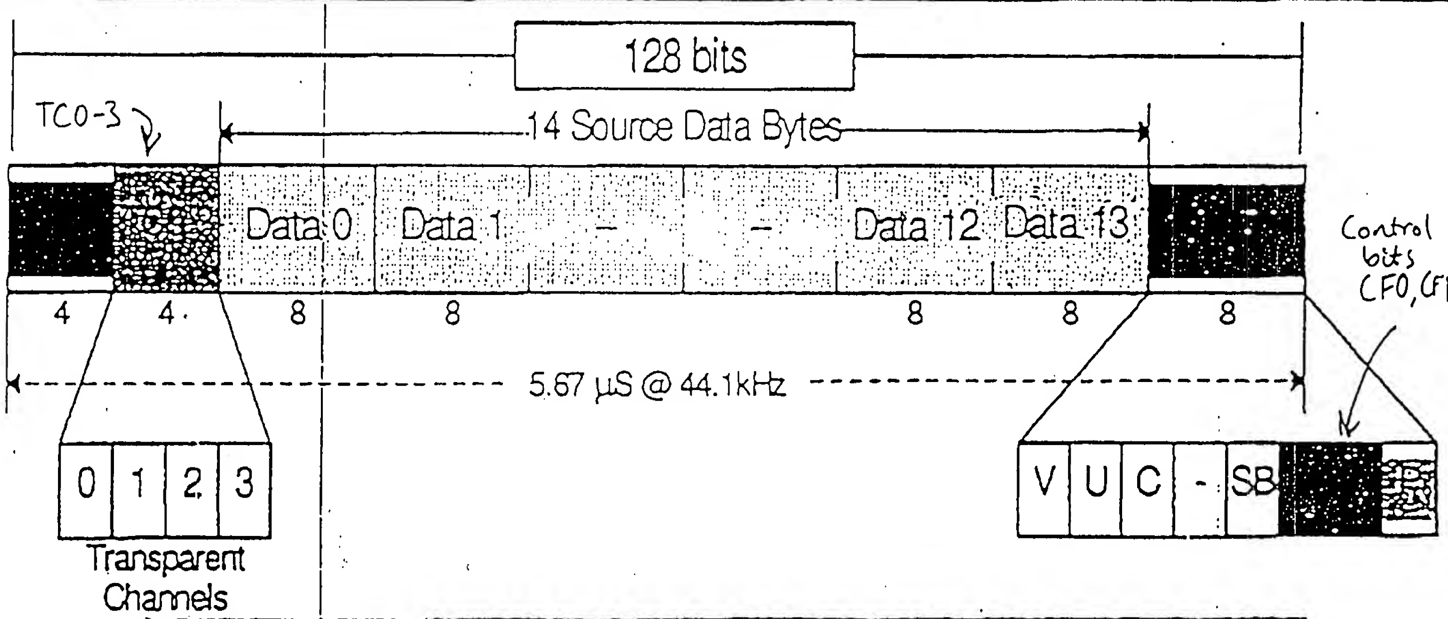
*DVD ROM Player must output compressed composite datastream (PES) to High Speed D2B Network (Max 11 MBPS)

High Speed D2B Frame Structure



In other systems sub-frame order and sizes can be different within each frame, and size of control frames can vary.

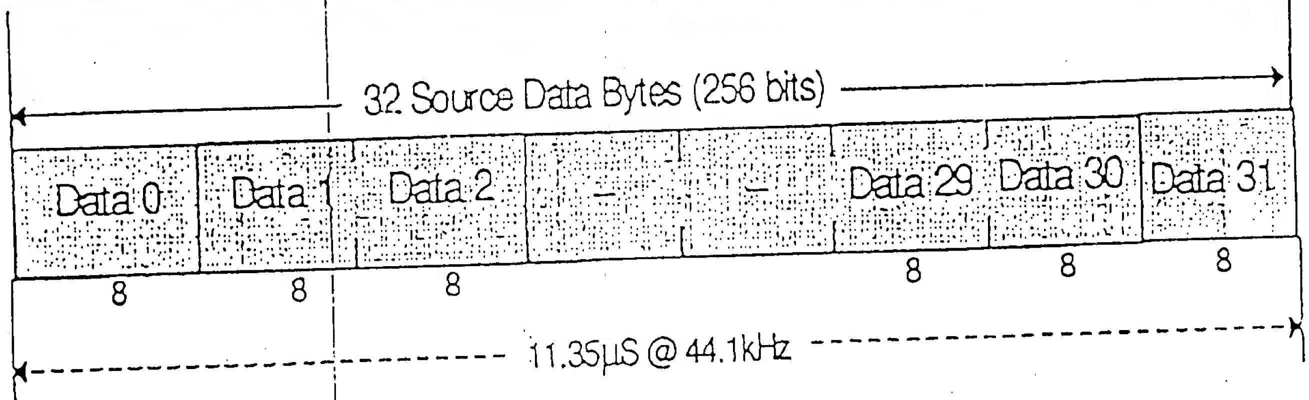
High Speed D2B Sub-frame Structure (Mode0)



Source data rate = 2*8*14*48 kHz = 10.752 MBPS
Multiple (up to 7) CD audio channels compatible with Super CONAN frame structure
two such sub-frames (Left and Right) per frame
Circuits switched connection i.e. capacity remains fixed throughout duration of connection

Mode0 Sub-frame same as "SuperCONAN", greater capacity than CONAN itself.

High Speed D2B Sub-frame Structure (Mode 1)



Packet based transport allowing mix of variable bit rates in quantities of 1 byte per subframe (equal to a source data rate of 384 kBPS at Fs = 48 kHz) up to a maximum of 32 bytes per sub-frame (equal to a source data rate of 12.288 MBPS)

Dynamic Allocation of source channel capacity in Mode 1 could be communicated using the control channel For example during thetransfer of a DVD image the required data rate may vary - thus a variable bit rate control protocol could be used to assign and de-assign more or less data bytes (1-32) on a multi-frame basis (ie a control frame comprise 48 source data frames and therefore the minimum time for modifying the Mode 1 source data rate is approx 1.09 msec

Alternativley a Fast Control Channel (FACCH) could be used comprising Data 0 (384 kbps) for capacity allocation

DVD Dynamic Rate Control

- Image resolution of 720 by 480 pixels encoded using 4:2:0 YCrCb video = 124.416 MBPS. Max compressed video rate is 9.8 MBPS.
- Minimum MPEG2 compression ratio for DVD is 12.7:1.
- Maximum of 41 kbyte per frame or 218 PES payloads each of 188 bytes. (1 PES payload every 160 usec)
- Inter-frame encoding drives compression therefore, max rate of change of data rate is once per frame - approx 30 msec
- Options for dynamic allocation of High Speed D2B Mode 1 capacity are :
  - Use control channel -30 control frames every video frame ? Problem with retry timeouts ? (Depends on traffic and arbitration)
  - Use Fast Associated Control Channel (FACCH) eg dedicated packet 1 (384 kbps) to allocate Mode 1 capacity

C&C Electronics

16

Further High Speed D2B Implementation Details:

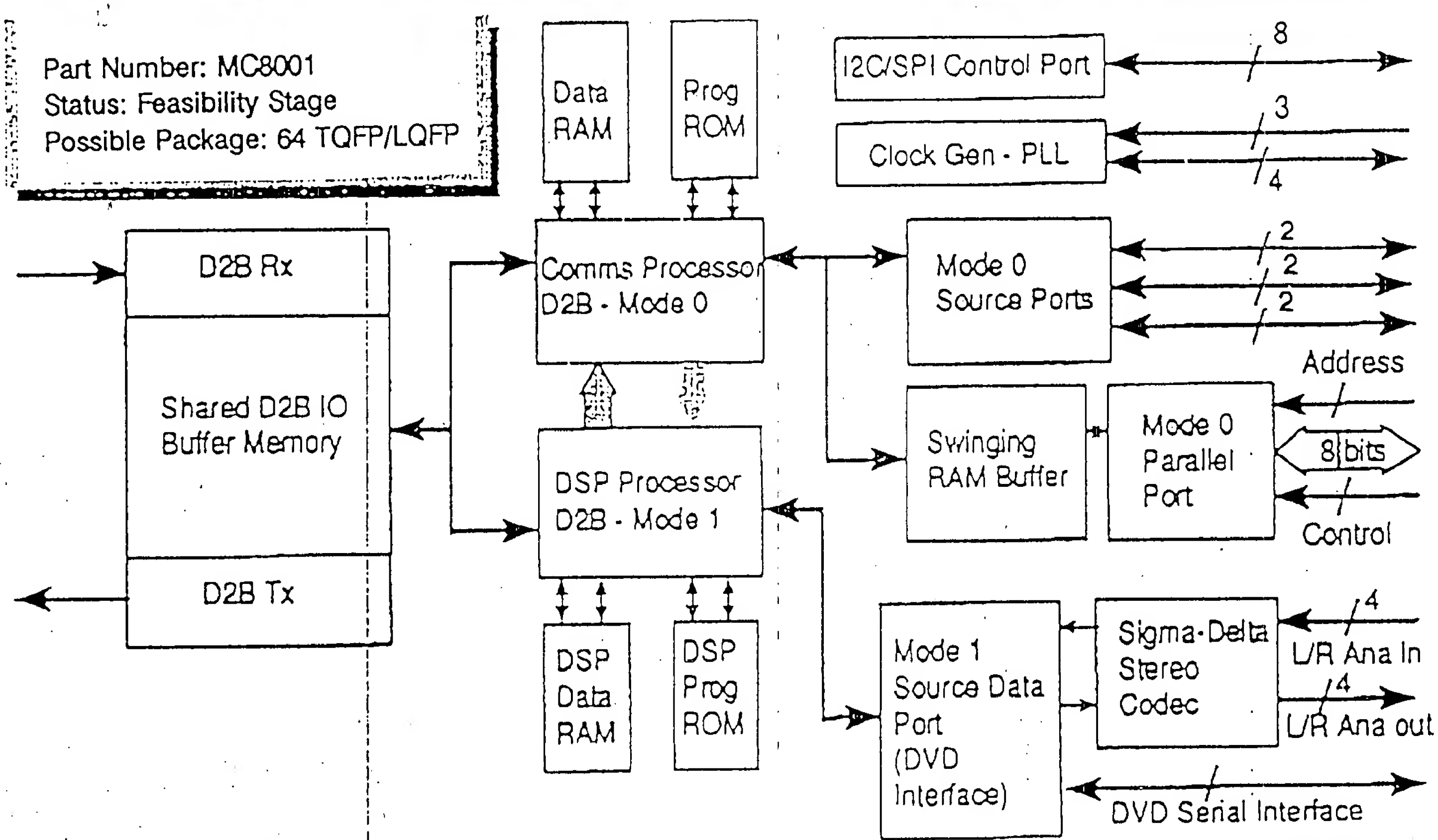
- Need to analyse and define control channel requirements in order to determine if more (or less) capacity than current 176 kbps is required
  - Construct simulation of typical D2B networks including DVD players and calculate control channel rate for alternative network proposals.
  - Repeat above exercise for packet based DVD data streams to determine how variation of DVD data rates affects Mode 1 data transfer and control channel capacity
- Decide on improved line coding approach -
  - eg 20% overhead using a 4B/5B code as used in FDDI networks
- Mode 1 Layer 2/3 capacity allocation could be defined in next step ?

"MegaCONAN" interface chip for High Speed D2B

- Implements all features of High Speed D2B
- Part Number MCI8001
- D2B Gross data rate in excess of 24.576 MBPS at Fs = 48 kHz and before line coding to FOT
- Two Simultaneous Communication modes :
  - Mode 0 : Circuit switched for CONAN and Super CONAN interface
  - Mode 1 : Packet switched for variable data transfer rate from 384 kbps to 12.288 MBPS at Fs = 48 kHz
- High performance integrated stereo audio CODEC
- Fully backwards compatible with CONAN and D2B Optical networks to allow multi-speed D2B networks to be constructed

"MegaCONAN" chip architecture

Part Number: MC8001
Status: Feasibility Stage
Possible Package: 64 TQFP/LQFP



Detailed Proposal II

The following pages present a modified embodiment, also providing both variable and fixed rate channels, but with flexible allocation of bytes within a common source data channel. Notable changes relative to Detailed Proposal I above are explained as follows:

- The delay at each network node in processing each frame of Proposal I would have been greater than 1 sub-frame when using a network containing both D2B Optical and High Speed D2B network nodes. Since a conventional D2B Optical network can only handle a maximum of 1 sub-frame delay around the network such a "mixed mode" network would not have functioned correctly. In the Proposal II the maximum delay is 12 High Speed D2B bits (= 3 D2B Optical bits) thus up to a theoretical limit of 20 nodes (16 when accounting for typical processing delays through each node) can be placed in a mixed mode network.

- In the Proposal I the allocation of circuit-switched synchronous traffic (Mode 0) and asynchronous packet based traffic (Mode 1) was fixed at 256 bits each. In Proposal II the traffic can be allocated in a flexible manner from 100% Mode 0 to 100% Mode 1 in increments of 1 source byte from 0 source bytes to 60 source bytes in a frame. Total source data capacity is now 23.04 MBPS at Fs = 48 kHz (DVD sample rate).

- The frame structure allows transmission of 1 complete ATM cell (53 bytes equivalent to a bit rate of 20.352 MBPS) in 1 frame as Mode 1 traffic

with 7 bytes left for Mode 0 traffic (2.688 MBPS or, for example, 2 stereo digital CD channels). Thus the network can be used to transparently connect nodes with ATM data interfaces, if desired.

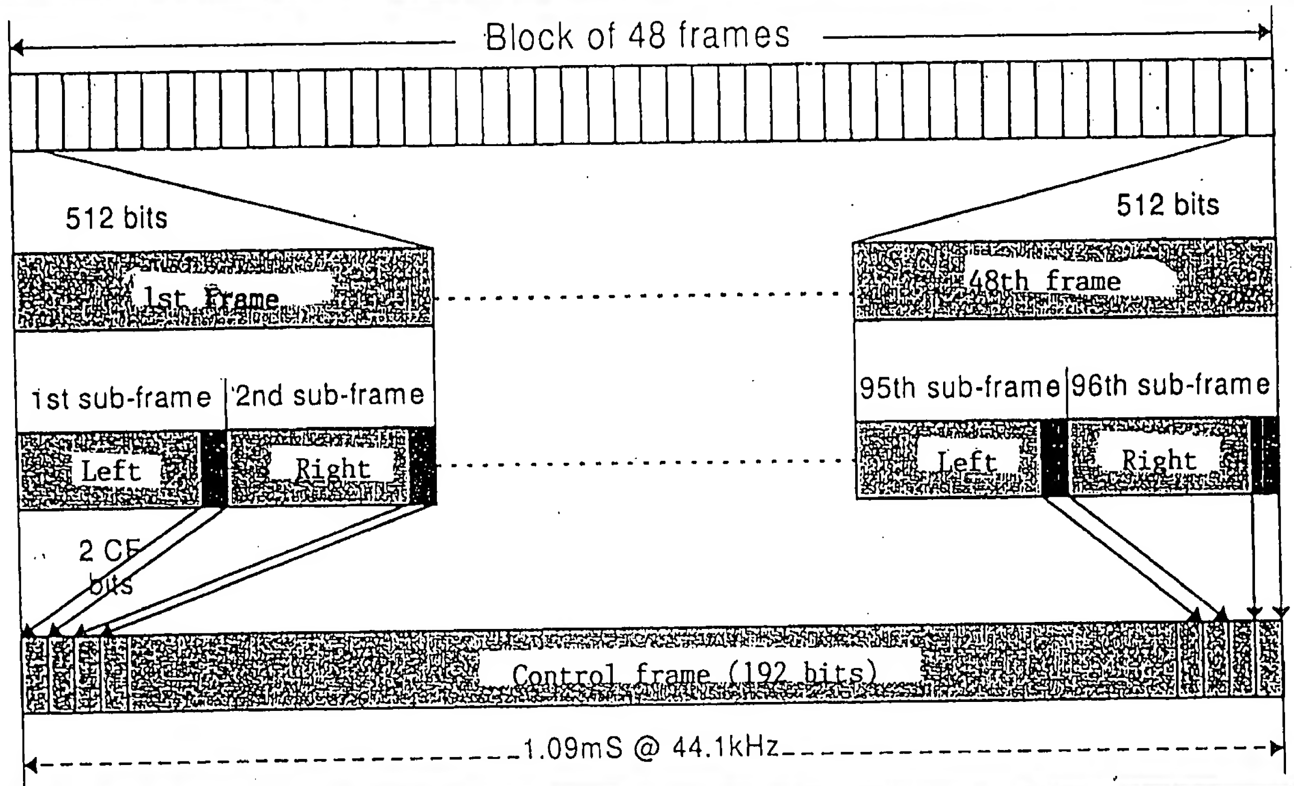
- Compatability is maintained as before by using a common control channel structure as currently used in D2B Optical sytems (at a rate of 176.4 kbps at Fs 44.1 kHz) ("Conan" technology).

- When using a network in which all nodes are High Speed D2B Nodes additional control channel capacity can be added by using the 4 bits in the Right Pre-amble to increase the control channel capacity to 352.8 kbps at Fs = 44.1 kHz.

- Line coding efficiency is improved using 4B/5B line coding to reduce the overhead to just 20%. Thus the rate at which optical transceivers are required to be driven reduces to 29.4912 MHz (at Fs = 48 kHz) as compared to 49.152 MHz for bi-phase encoding.

Statistical multiplexing can be used to multiplex up to 4 DVD channels onto the High Speed D2B Bus. This relates to calculating node buffer sizes in a distributed video transmission network.

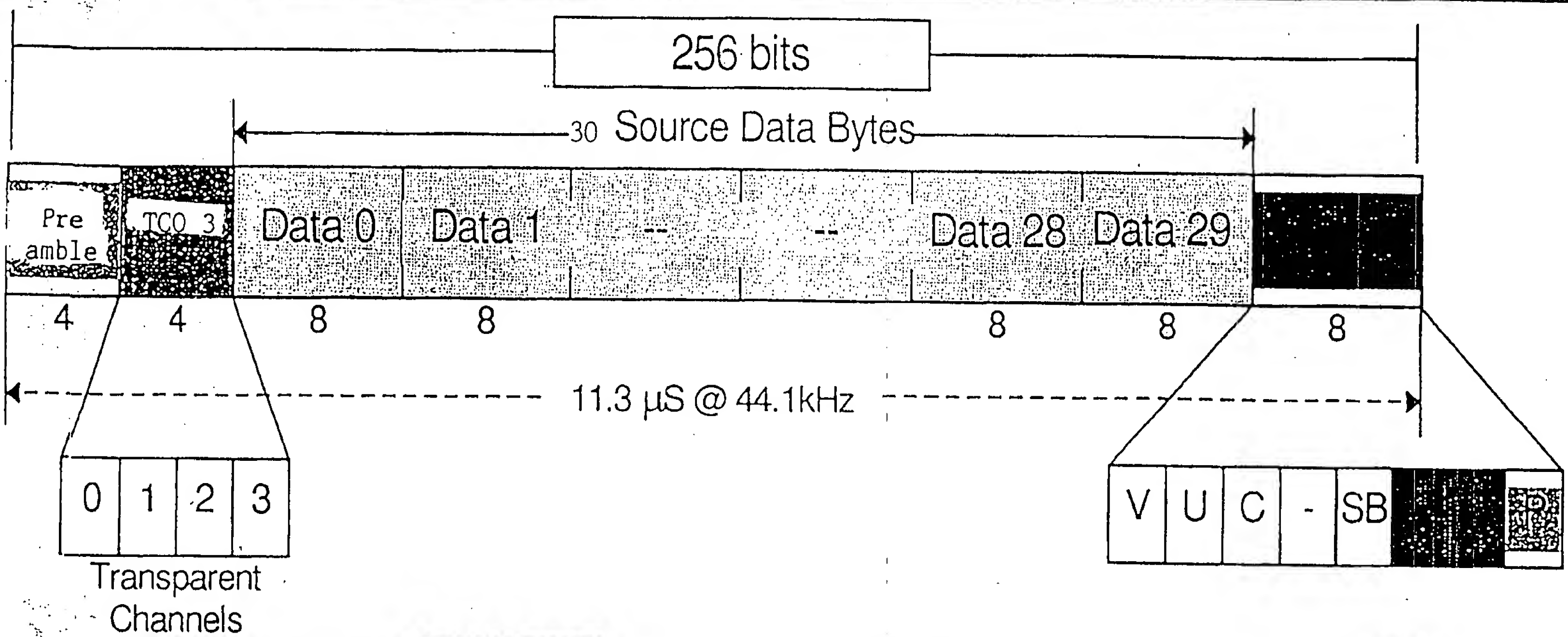
High Speed D2B Frame Structure



Mixed Mode 0 and Mode 1 data in common source channel.

Gross bit-rate 512 x 48 = 24.576 MBPS

High Speed D2B Sub-Frame Structure



Total Source data rate = 2*8*30*48 kHz = 23.04 MBPS
Synchronous (Mode 0) capacity allocated from frame start
Asynchronous (Mode 1) capacity allocated from frame end
Two such sub-frames (Left and Right) per frame
Right pre-amble (4 bits) can be allocated to control channel for High Speed D2B only networks (384 kbps at Fs = 48 kHz)

C&C ELECTRONICS

High Speed D2B Features

- Frame structure designed for compatibility with D2B Optical and D2B Optical Plus
  - Maximum Delay through each mode is 12 HSB bits = 3 D2B bits when in using mix of High Speed and D2B Nodes
- Asynchrononous traffic alocated in source bytes from the end of frame - Synchronous traffic from the beginning if the frame.
- Control channel format is common to D2B Optical and D2B Optical Plus
- Additional control channel capacity may be used when all nodes are High Speed D2B nodes
  - Optional use of right pre-amble to increase control capacity to 384 kBPS at Fs = 48 kHz

Detailed Proposal III

A further example will now be presented, which differs from Detailed Proposal II in various ways.

The number of source data bytes per sub-frame is increased to 46, giving a continuously allocatable 92 source data bytes per frame. The frame rate is fixed at 48 kHz, giving a higher overall data rate than in Detailed Proposal II.

Detailed Proposal III also provides more detail of the control of asynchronous channel allocation, and a package structure for data within the asynchronous channels.

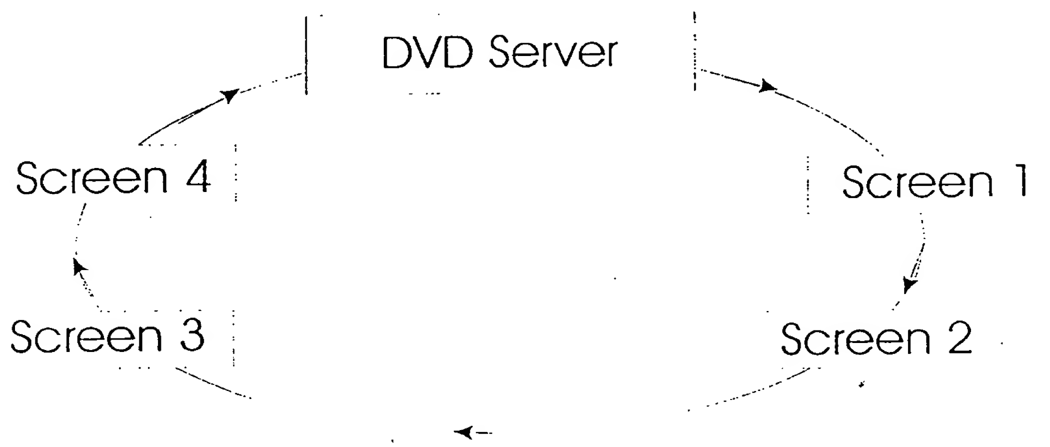
Although the variable width (asynchronous) channels and the fixed rate (synchronous) channels are again allocated within a single source data field from different ends, in this proposal the asynchronous traffic is allocated in the source bytes from the beginning of the frame, not the end. At the start of each block of 48 frames, asynchronous block headers are provided which indicate a channel ID and channel width which are fixed for the remainder of that block. The header for successive channels is found by counting through the source data bytes of the first frame in accordance with the width of each channel. The synchronous data channels are allocated from the end of the source data field.

Packets carrying 42 bytes of source data in this example can also be grouped into packs of up to 256 packets. This can assist data handling in applications where larger segments of data, such as disk segments of 2 kbytes are expected. A DVD source, for example, normally provides data in so-called PES cells of 188 bytes, which could, if desired, be grouped as pack of 5 of the proposed asynchronous data packets.

A detailed description of Proposal III now follows.

System Description

A High Speed D2B Optical system consists of a set of devices which are connected in a ring topology via a series of point to point links. Each of these links are physically independent.



Example High Speed D2B System

Depending on its function, each Device in the system can:

- supply, receive or pass-through source data (e.g. digital audio, video etc.).
- send and receive control messages

To support the sending and receiving control messages, each device has two unique addresses an application-related address and a ring-position related address. It is also possible to broadcast a control message to all devices or to a pre-selected group of devices.

The protocols for control message communication are defined in the *D2B Optical Specifications*.

HS D2B Performance

At a frame rate of 48 kHz the High Speed D2B System offers a gross data rate of 36.864 Mbps and a net source data rate of 34.56 Mbps (organised as 92 source bytes per High Speed D2B Frame).

High Speed D2B Frame Structure

The frame and sub-frame structures for High speed D2B are shown in Figures 1 and 2 respectively.

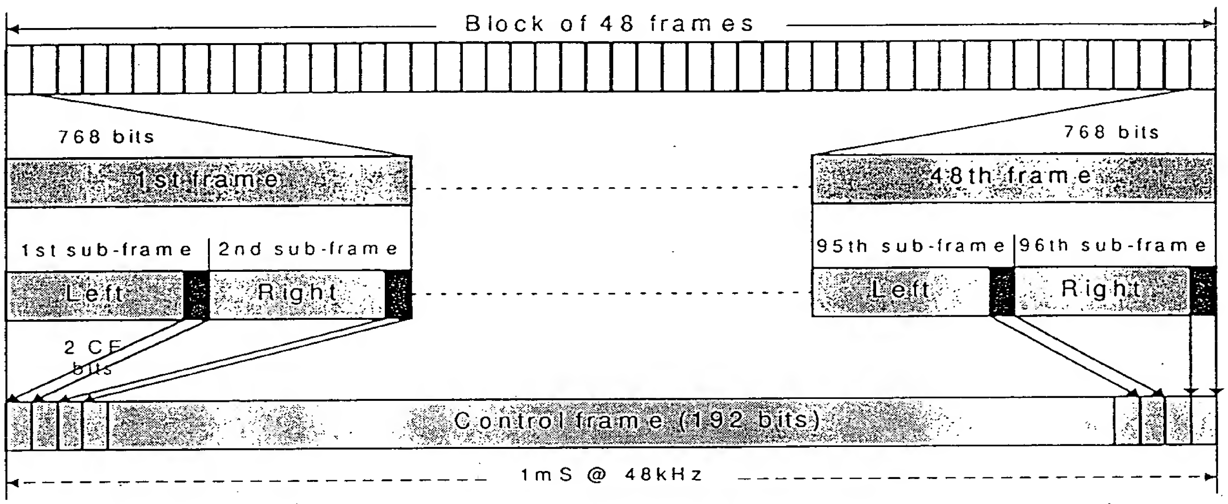
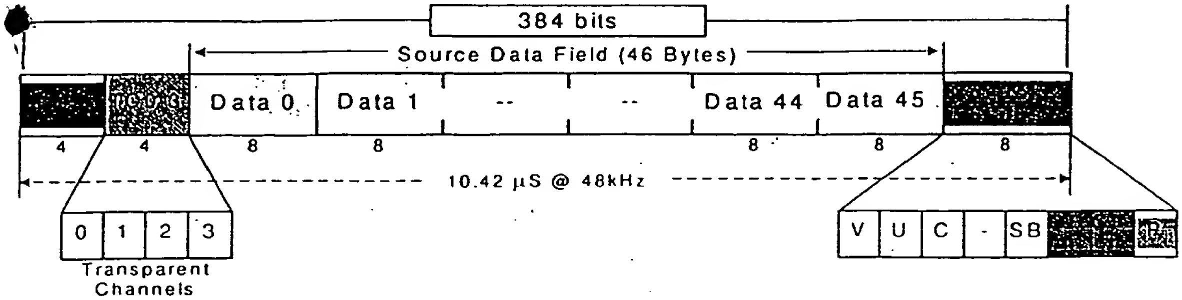


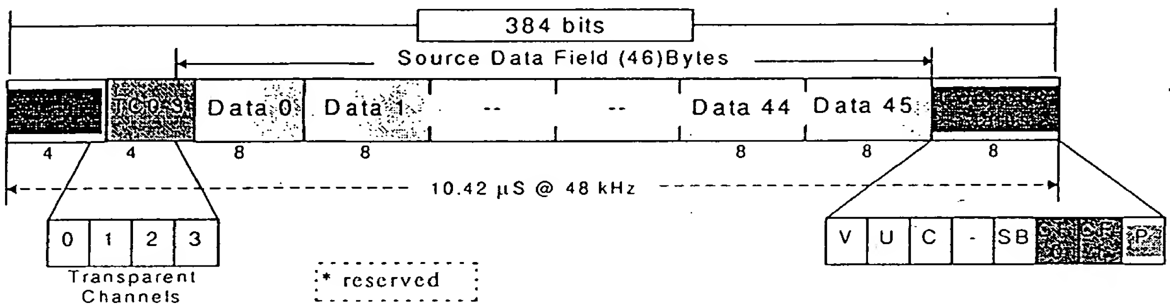
Fig 1 : High Speed D2B Frame Structure

The High Speed frame consists of two identical sub-frames, to provide some compatibility with the earlier CONAN system as in Proposal II, but this is not essential. The frame rate is proposed to be fixed at 48 kHz. Synchronous channels requiring a lower rate (e.g. CD audio, MPEG 1) can be padded and buffered accordingly within the synchronous data channels.

First Subframe



Second Subframe



In other embodiments, the number of bits in the frame and hence the number of bytes in the source data field may be different.

Error Protection

The HS frame is protected by 2 parity bits, 1 in each subframe.

Source Data Transport

Whenever source data (e.g. digital audio or video) needs to be transported over HS D2B, a source data connection must be established. This is called *connection set-up*. During the set-up, the required number of source data channels (bytes) are allocated from free channels within the HS D2B. For example, to carry a stereo audio signal from a CD player requires an allocation of 4 bytes. Source Data Connection protocols based on control messages are used for setting-up and removing connections. *

For synchronous connections, this capacity remains allocated until the connection is removed. Synchronous connections have no superimposed framing or packet structure , although applications are free to provide structure as desired.

For asynchronous connections, the connection set-up establishes the starting allocation. However this allocation can be varied during the lifetime of the connection as described in the section on Asynchronous Connection Blocks.

When all the capacity has been allocated, attempts to build further connections will fail. When this happens, the controlling AVC must decide which existing connection(s) (synchronous or asynchronous) need to be removed to release enough capacity for the new connection. The complexity of the allocation is hidden from the controlling AVC since each device is responsible for managing the allocation in its own output link (ring segment).

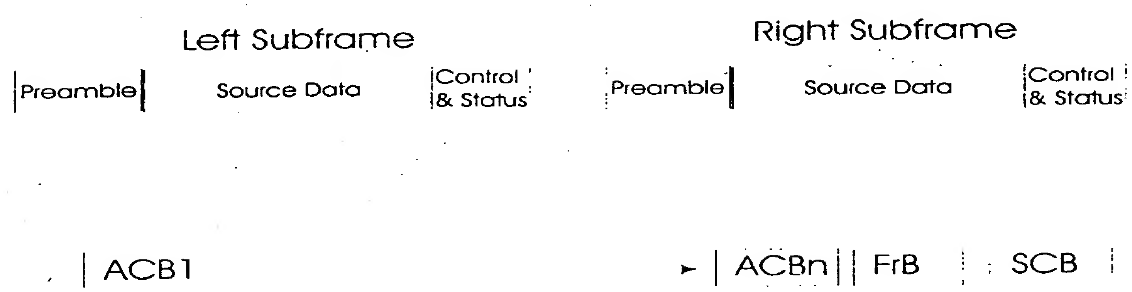
Allocation of Source Data Capacity

The source data field comprises the source data fields of the two sub-frames (with 46 + 46 bytes capacity), flexibly partitioned into variable size sections as shown in the diagram.

The first part is allocated to variable-rate asynchronous transport: whilst the synchronous (or fixed-rate asynchronous) source data capacity is allocated starting from the end of the frame.

* Source data routing is similar to that of the CONAN IC, but with a larger number of bytes per frame, and hence a far greater number of switching permutations. Connection building can be performed for example by protocols based on the disclosure of EP-A-0360338 (PHN 12678) and EP-A-0432316 (PHN 13189), adapted according to the ring topology protocols for this purpose are established using the control message frames to carry pre-arranged connection request instructions.

Source Data Field Structure



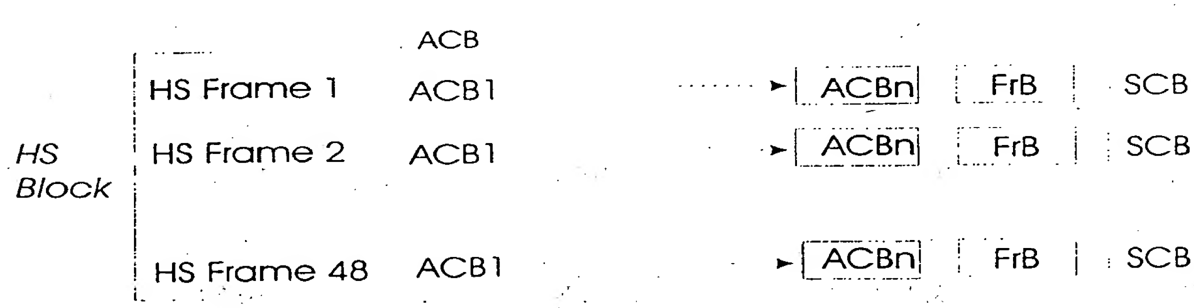
1. **ACB1..ACBn** are Asynchronous data Connection Blocks — A bytes (variable)
2. **FrB** represents free capacity for asynch (or sync) conns. — 92 - (A+S) bytes
3. **SCB** represents the synchronous data connection block — S bytes (variable)

Asynchronous Connection Blocks (ACB)

Asynchronous connection blocks are the means by which multiple variable-rate source data connections can be carried on HS D2B. They are the containers for asynchronous connections within the HS frame, carrying the packet switched data. Since more than one ACB may be present in the same frame allowing multiple simultaneous asynchronous connections.

The ACB is segmented over a block of 48 HS frames (aligned to the block of 48 frames used for transporting control message frames, see figure 1). Note that the ACB header appears only in the first frame of the block of HS frames. The size of the ACB is: ACB width * 48 bytes. Thus by varying the width of the ACB from block to block, the capacity occupied by an asynchronous connection can be varied, subject to the limit of the total capacity of the frame.

Asynchronous Connection Block (ACB)

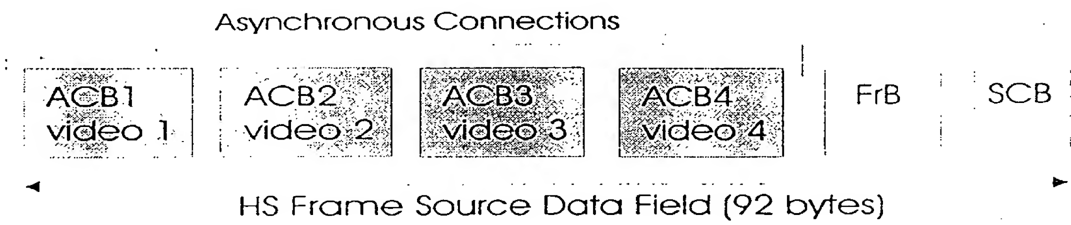


Example Application

The system shown on the first page consisting of a DVD server sourcing 4 different video signals could make use of the following source data field structure. The bit rate allocation may be varied for each connection as described in the section on Asynchronous Connection

Blocks. Note that since the destinations for the video signals are distributed around the system, not all asynchronous connections need to be present in all links in the system. For example, the asynchronous connection carry the video signal to Screen 1 (video 1) needs only to be present in the link from the DVD server to Screen 1. Each of the asynchronous connections could have a starting width of 24 bytes (per frame) and then could be varied individually as required for the variable rate video signal.

Source Field Data Allocation for Multiple Video Sources



Error Protection

The contents of the Asynchronous Data Connection Blocks rely on protection within packets.

Each Asynchronous Connection Block (ACB) is structured as follows:

Asynchronous Connection Block (ACB)

ACB Header	ACB Data

ACB-Header

ACB ID	6 bits
Start of Packet flag	1 bit
Reserved	1 bit
ACB width	7 bits
Reserved	1 bit

Notes

1. The *ACB ID* enables a receiving device to identify the connection whose data is carried by this block.

2. The Start of Packet flag indicates whether the first data byte of this ACB is also the first byte of a packet (flag set to 1) or whether it is a continuation of a packet. This allows for longer packets than the type detailed below, for example.

3. The Reserved fields is for future extensions.

4. The ACB width field indicates the number of (consecutive) bytes allocated to this asynchronous connection within each frame, encoded such that 1 means 2 bytes, 2 means 3 bytes etc. The minimum width of two bytes ensures space for the header in the first frame of the block. The ACB width may be restricted to ensure an integral number of packets within a block, where packet and/or frame sizes vary from these examples.

ACB-Data

Within the capacity provided by the ACBs, source data is carried in the form of packets. The packet format is described in the following section.

The ACB header format and its field sizes can be different according to the application.

Free Capacity (FrB)

The free capacity is held within an Asynchronous Connection Block (ACB) with ID = 0. This allows the hardware to identify the synchronous connection blocks easily.

Synchronous Connection Block

This block van be used to carry both synchronous signals e.g. 16 bit PCM audio at 48 kHz or asynchronous signals whose bit-rate is fixed. Changes to the contents and size of this block can only be made by setting up a new connection or removing an old connection. These operations are defined in the source data connection protocols [1].

Packet Structure

Asynchronous Data carried within either asynchronous or synchronous connections is formatted into packets whose structure is described below. This provides framing to allow a device receiving the data to identify the data and recover it correctly. Since each packet has its own ID, it is possible to interleave different streams of data over the same connection. For example a particular connection might carry predominantly packets containing video data interleaved with an occasional packet for control purposes.

Packet Format

Packet Header	Data	CRC

Packet Header

Packet Type	2 bits (the remainder of the packet definition applies for type 0)
Packet ID	3 bits
Reserved	1 bit
Flow Control	1 bit
Start of Pack	1 bit
Remaining Packets	8 bits
Number of bytes used.	8 bits

Packet Data

Data	42 bytes

Error Protection

Checksum/CRC	1 byte

Notes on Packet Header

1. Packet Type identifies the format of the packet, for example longer packets may be provided for bulk data transfer, as opposed to real-time channels.

2. Packet ID identifies the type of data contained in the packet, such as audio/video/general data, to assist routing in the destination device. Packet ID "7"H is reserved for control (e.g. connection management) messages, with low latency compared with the existing control message channel (CF bits).
3. Flow Control is used by a receiver of the data to indicate that its Rx buffer is full (when this flag is set to 1). When this is detected by the source of the data, it will normally suspend transmission.
4. *Remaining Packets* indicates the number of packets remaining within the current pack (group of packets)
5. *Number of Bytes* used indicates the number of bytes in this packet containing valid data

Flow Control

The flow control mechanism implemented via the flag in the packet header, requires there to be a connection from the destination device back to the source device. This connection, which would be built as part of the connection set-up of the signal whose flow is being controlled can have a much reduced capacity (minimum 1 byte per frame) compared with e.g. the video signal to which it refers. It may for example have the same ACB ID, and use the Packet Header format. A single byte channel Could also be allocated as an SCB.

Alignment of a packet within an Asynchronous Connection Block

The start of the packet is indicated by the Start of Packet bit in the ACB header. When this bit is set, the first byte of data following the ACB header is also the first byte of a packet. When this bit is not set, it indicates that the contents of the ACB are a continuation of a previous packet.

Segmentation of the Packet

The number of HS frames required for transmission of a packet is a function of the size of the packet and the width of its containing ACB in each HS Frame. If the ACB is *n* bytes wide then the packet will encompass (packet_size + size of ACB Header)/ 48n HS frames.

Diagram showing how packets are loaded into an ACB

The diagram below showing how an ACB of width 6 bytes is loaded with packets (of size 46 bytes). This ACB holds six packets of which the first two are shown (A and B). Note that the ACB header occupies the first two bytes and that between each packet are 2 reserved bytes which are used as padding. Each ACB occupies the next available group of bytes, according to the specified width of each channel. If a connection becomes wider or narrower at the next ACB, the ACB for other connections are shifted up or down accordingly.

- Packets carried in an Asynchronous Connection Block (ACB)

	ACBn					
Fr. 1	ACB_H	ACB_H	pA1	pA2	pA3	pA4
Fr. 2	pA5	pA6	pA7	pA8	pA9	pA10
Fr. 8	pA41	pA42	pA43	pA44	pA45	pA46
Fr. 9	Res.	Res.	pB1	pB2	pB3	pB4
Fr. 10	pB5	pB6	pB7	pB8	pB9	pB10
Fr. 16	pB41	pB42	pB43	pB44	pB45	pB46

Key
Fr. .... Frame
ACB_H ..... ACB Header
Res. .... Reserved
PA1 .... Packet A, first byte

Packet Buffers

Each HS D2B device which needs to send or receive asynchronous data will require buffers for packets which have been received or are to be sent. The size of these buffers will be defined according to requirements.

APPENDIX - CONAN IC DATA SHEET PAGES 5-32

5

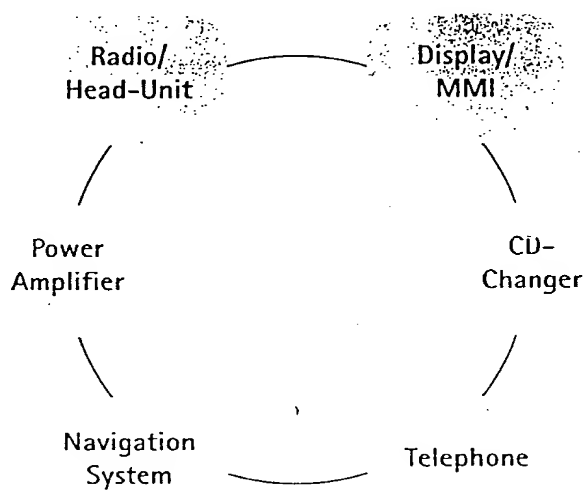
1. Introduction

1.1 Overview

The Conan technology is a leap forward in the development of Communication and Information networks for automotive consumer electronics products. It combines Source Data (digital audio, video or any other high-volume, real-time digital data) and Control Data, with the advantages of the Plastic Optical Fibre medium (i.e. high data capacity and immunity to interference and noise).

To achieve effective network design and facilitate product implementation, the OCC8001 Optical Transceiver ('Conan') provides maximum functional integration, including source data routing, communication management and all communication protocol tasks.

Conan supports network protocol standards such as D2B Optical, and is fully compatible with systems built on a D2B Optical network.



Example System with a Ring Topology

The data throughput on a Conan network is impressive. At a sampling rate f_s of 44.1 kHz, it offers:

Source Data	4.2 Mbps (equivalent to 3 x 16-bit stereo audio channels).
Control Data	Up to 176.4 kbps (918 control frames per sec).
4 Transparent Channels	Up to 88.2 kbps each.

1.2 Features

Conan has been optimised for implementation in consumer electronics products, and offers these features and benefits:

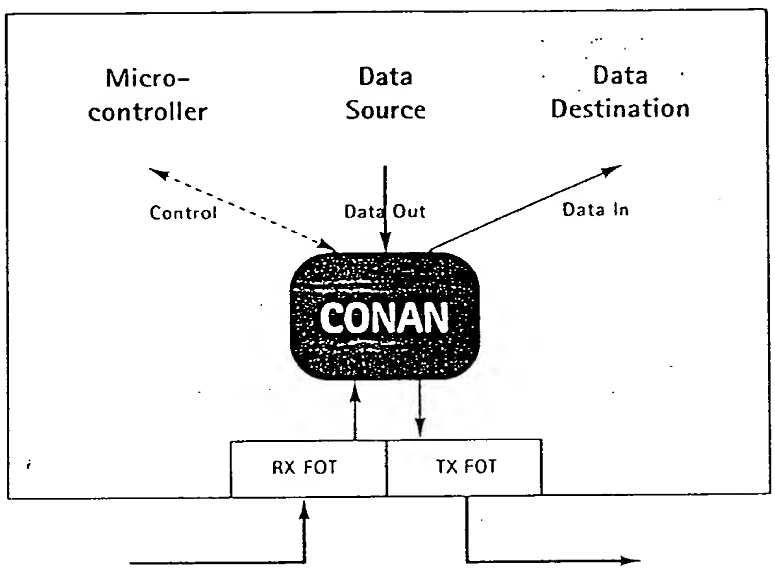
Features	Benefits
Integration of Source & Control Data	Reduces the number of components required for signal distribution and control
Multiple Source Data Channels	Allows simultaneous bi-directional transmission of multiple source data channels
Multiple Source Data Ports	Only one Conan needed per product, even with several internal sources or destinations.
Multiple Source Data Formats	Adaptability to a variety of components which support different digital data formats
SPDIF Audio Port	Provides easy integration with existing digital audio products
I2C- and SPI-compatible Control Ports	Easy interface to wide range of microprocessors
Flexible Source Data Routing	Versatile use of the network capacity
Transparent Channels	Easy implementation of proprietary control messaging
Programmable Clock Manager	Easy integration in different products
Flexible Synchronisation	Network clock may be derived from a variety of sources
Low Jitter PLL	Low jitter ensures high audio quality (i.e. low THD and low noise)
Fail-safe mechanism (Electrical Bypass)	In case of device failure, network operation can be maintained
Network Position Identification	Each device can get its position on the ring
Versatile Control Messaging	Different addressing methods (incl. broadcast) make best use of control channel capacity
Automotive Temperature Range	-40°C to +85°C (contact C&C for wider ranges)

1.3 Applications

Conan has been specifically designed for easy implementation and integration of networked audio, video and communication products. Conan is optimised for use with (but not limited to) a ring topology, based on plastic optical fibre, for automotive applications, though can be equally well employed on other media (e.g. copper) and for other application areas.

Products at nodes on the network can be sources or destinations for source data (or both, or neither), for example, CD-Changer (source), Amplifier (destination), Radio/Head-Unit (source and destination).

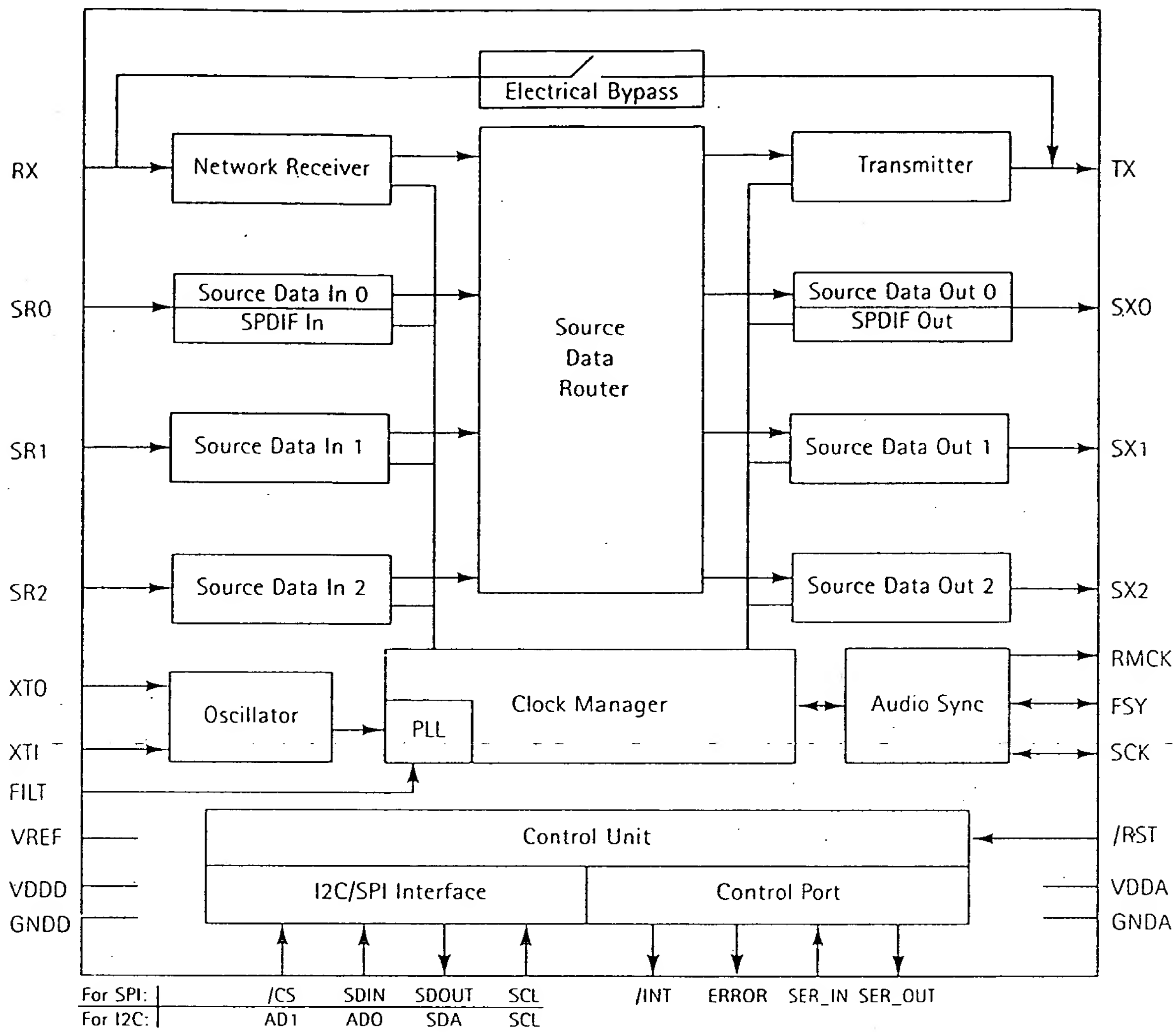
A product can generate source data for transport on the network, and/or retrieve source data from the network. It can additionally send or receive control messages. Conan may be incorporated into a product as shown below.



A Conan Device on the Network

To achieve the highest possible efficiency, the network timing is synchronised to a System Master. The System Master is a product on the network which is able to generate the timing for the entire network. Only one Master device may be active on the network at any time. All other devices are Slaves. The Slaves get their timing from the Master by recovering the clock from the bit-stream received from the network. Conan can be configured by software as either a Master or a Slave. Up to 24 nodes may be present on the network.

1.4 Block Diagram



Conan Block Diagram

Conan contains a Network Transceiver (Rx and Tx) to interface to the network. The network receiver recovers the clock and decodes the incoming bit-stream. The network transmitter encodes the outgoing data and transmits it on the network. Conan can be configured by software as either a Master or a Slave node on the network.

Conan provides flexible access to up to 12 source data bytes within one frame (e.g. configured as three 16-bit stereo channels) on the network. The three source data input ports (SR0, 1 & 2) and three source data output ports (SX0, 1 & 2) can transfer 8, 16, 24 or 32 bit source data into/out of the Conan through the Source Data Input and Source Data Output registers.

The Router provides an easy and flexible way of controlling the routing of source data. All source data bytes, coming either from the network receiver or from the source data input ports can be re-arranged, mixed and re-directed to the network transmitter or the source data output ports. Conan can also support SPDIF (also known as AES/EBU or IEC-958). Conan can even be used to route source data locally, from I.C. to I.C, within a product.

In Slave mode, Conan's timing is derived from the incoming bit-stream. The network receiver recovers a clock from the incoming bit-stream using a low jitter, clock multiplying, phase locked loop (PLL). The PLL will lock to incoming bit-streams with sample rates of 30 to 50kHz. On loss of lock, the PLL is pulled to its lowest frequency (to minimise EMI).

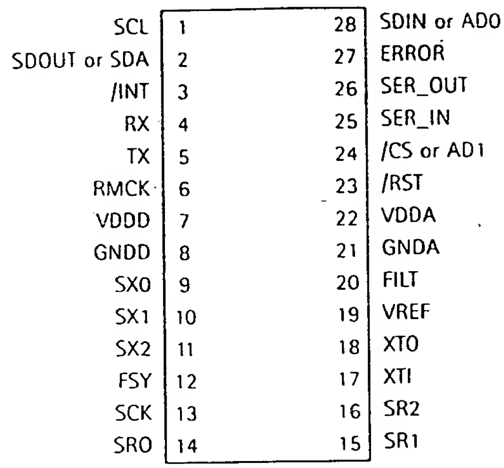
In Master mode, Conan's timing may be derived from a crystal, a source data clock applied to SCK or an SPDIF channel applied to SR0. The Oscillator circuitry allows a crystal to be connected directly to XTI/XTO, or some other clock source to be used. The PLL multiplies-up this timing source. The RX pin is over-sampled by a high-frequency clock and data is recovered by a digital state machine.

An Electrical Bypass between RX and TX pins allows a node to be bypassed electrically. This bypass is automatically activated on power-up or reset as a safety feature. It helps to reduce the network start-up time, and can also be used as a fail-safe to isolate the node in case of loss of lock or other product failure (typically triggered by a watch-dog function).

The Control Unit provides the interface from Conan to a microprocessor. It provides an I2C- or SPI-compatible interface, interrupt and error outputs, and transparent channel input/output. Conan is a slave on I2C/SPI, so transfers must always be initiated by an external master.

Internal registers are used to buffer the source data and control messages. It also contains the Routing Information Table (RIT) and control and status registers. These registers are accessible for reading or writing through the I2C/SPI interface.

1.5 Package Pin-Out and Pin Descriptions



Pin	No.	I/O	Function	Pin	No	I/O	Function
SCL	1	In	I2C & SPI clock	SDIN or AD0	28	In	SPI data in or I2C address select
SDOUT or SDA	2	Out	SPI data out or I2C data	ERROR	27	Out	Error indicator
/INT	3	Out	Interrupt output (open drain)	SER_ OUT	26	Out	Transparent channel out
RX	4	In	Network receive	SER_ IN	25	In	Transparent channel in
TX	5	Out	Network transmit	/CS or AD1	24	In	SPI chip select or I2C address select
RMCK	6	Out	Received master clock	/RST	23	In	Reset (active low)
VDDD	7	-	Power rail	VDDA	22	-	Power rail
GNDD	8	-	Ground	GNDA	21	-	Ground
SX0	9	Out	Source data output port 0	FILT	20	-	Loop filter
SX1	10	Out	Source data output port 1	VREF	19	In	PLL voltage (and reset timing)
SX2	11	Out	Source data output port 2	XTO	18	Out	Crystal out
FSY	12	I/O	Frame sync.	XTI	17	In	Crystal in
SCK	13	I/O	Shift clock	SR2	16	In	Source data input port 2
SR0	14	In	Source data input port 0	SR1	15	In	Source data input port 1

Pin Descriptions

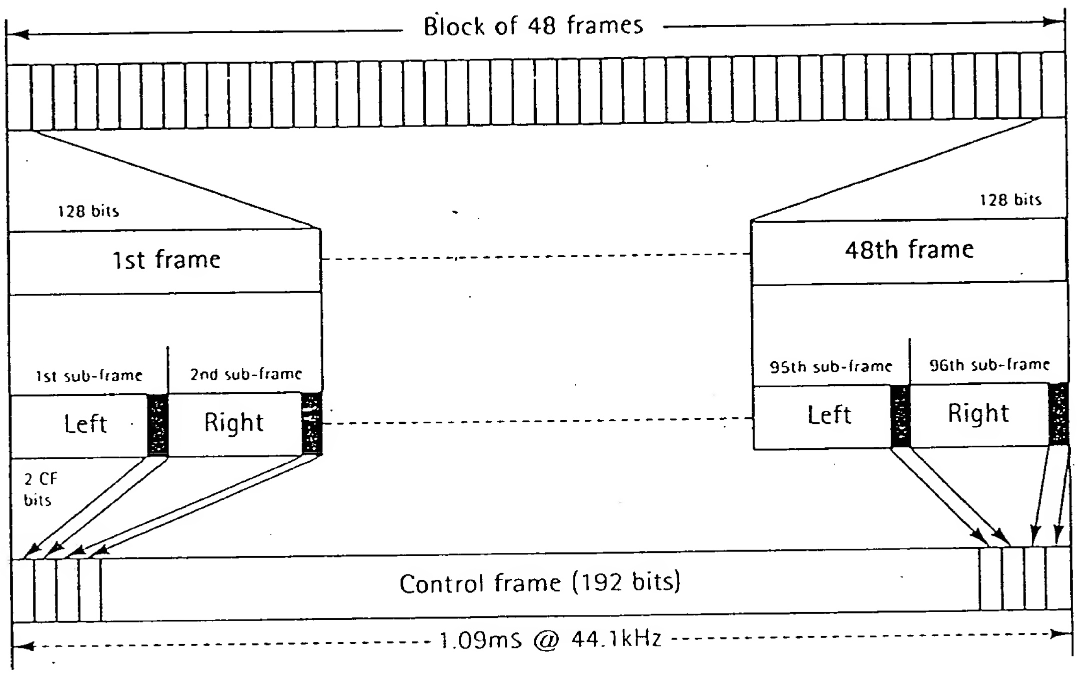
All pins are CMOS TTL logic level, except power, ground, FILT, VREF, XTI and XTO.

2. Functional Description

2.1 Data Transport

The source data and control messages are transported on the network from node-to-node in *frames* generated by the System Master. Frames are circulated at the same rate as the system sampling frequency, typically f_s =44.1kHz. Frames are grouped into *blocks* of 48 frames.

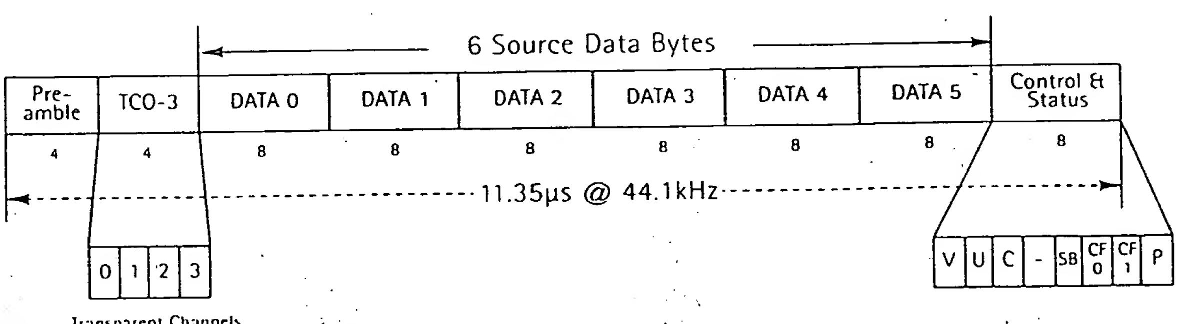
The frame is divided into two *sub-frames* ('left' and 'right'). At f_s =44.1kHz, there will be 88,200 sub-frames per second. The left sub-frame is always the first of the pair transmitted on the network. At the physical level, bits are transported with bi-phase encoding. The relationship between the block, frame, sub-frame and control frame is shown below.



Block, Frame, Sub-frame and Control Frame Relationship

2.1.1 Conan Sub-Frame

Each sub-frame contains 64 bits, handled within Conan as 8 byte fields. The fields comprise the preamble, the transparent channels, 6 bytes of source data, and 8 control/status bits which make up the control frames and the SPDIF status bits. The sub-frame is shown below.



Conan Sub-frame

Preamble

The preamble synchronises the network receiver. There are three types of preamble, identical to those defined in the IEC958 specification. They contain bi-phase coding violations which the receiver can recognise. The three unique preambles identify left, right and block sub-frames. The left preamble identifies the beginning of a frame and the block preamble identifies the beginning of a block. The block preamble replaces every 48th left preamble. This provides a block structure to which the control frame data is synchronised.

Transparent Channels

The four TC bits enable the transport of four serial channels for proprietary control or status information on the network, with no additional hardware or software overhead. The use of these channels is left open to System Integrators, who must define their own protocols for applications. Typical applications include the transport of raw control or status information, such as RS232-type data, VICS data, GSM data, PIN Card data, etc.

Source Data Bytes

The source data bytes carry the high-volume real-time digital source data. The twelve bytes per frame may be allocated flexibly, so that the devices in a system may use the source data bytes in the most efficient way for that system. The mechanism used for allocating the bytes is described in section 2.4, Source Data Routing.

Status Bits

If an SPDIF channel is being transported, the V, U and C bits contain the validity, user and channel status bits of the SPDIF channel. The left/right convention of these bits is determined by the left/right preambles of the Conan sub-frame.

The Start Block bit SB identifies the block boundary of a synchronous SPDIF channel and is set after every block of 192 frames (synchronised with the SPDIF signal that is being transported). This synchronisation is performed automatically by the chip. The Parity bit P generates even parity for the entire sub-frame.

Control Bits

The control bits CF0 & 1 carry the control messages (for controlling devices and sending status information). There are 2 CF bits per sub-frame, and a control frame is 192 bits long, therefore 96 sub-frames (48 left + 48 right) are required to build-up a complete control frame. The control frame is described in the next section.

2.1.2 Control Frame

The control frame is assembled from and aligned with a block of 96 sub-frames, i.e. the first two bits of a new control frame are taken from the sub-frame with a block preamble, and subsequent pairs of bits are taken from subsequent sub-frames to build up a control frame.

Arb	Destination Address	Source Address	Msg type	Msg Len	Data 0	----	Data 15	CRC	ACK	NAK	Reserved
2	12	12	4	4	8		8	16	2	2	10

← 1.09mS @ 44.1 kHz →

Control Frame

The fields of the control frame are:

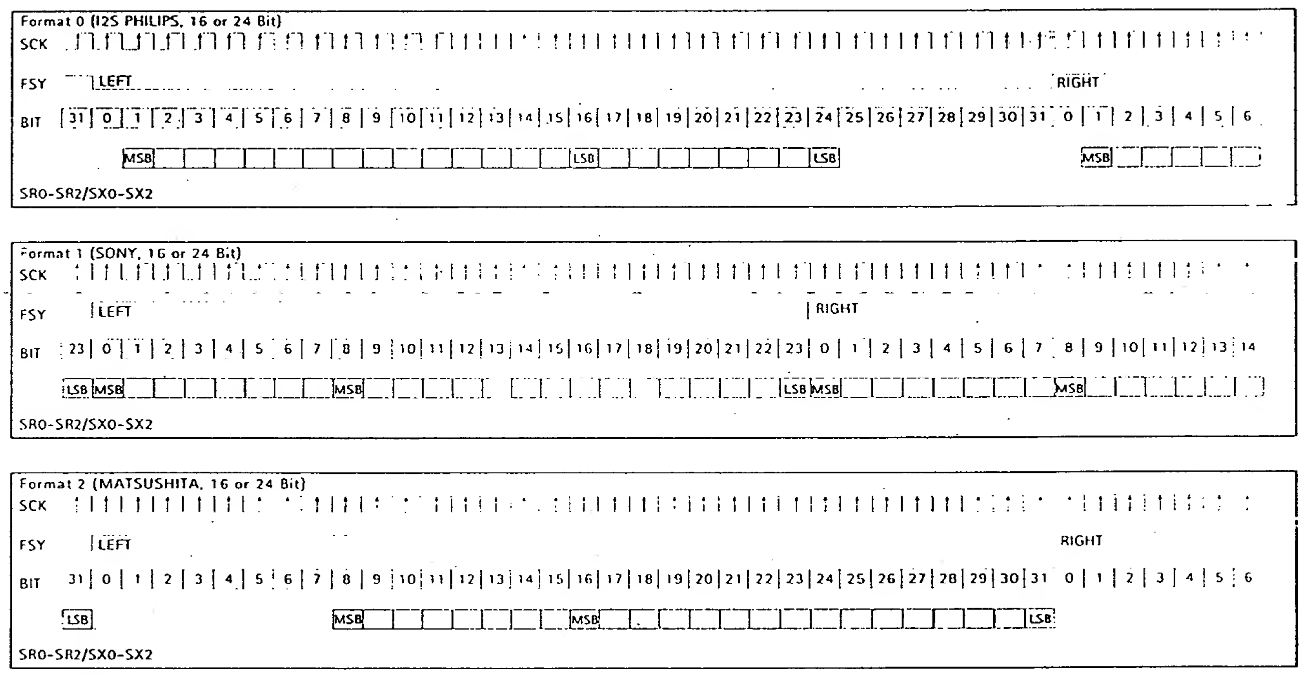
- Arbitration bits :
  These indicate if the control frame is free or occupied. Conan handles these bits automatically.
- Destination Address
  This is the 12-bit device address of the destination of the message, in the range '000'H to 'FFF'H. The sending device writes this into its message transmit buffer for transmission. Certain addresses and address ranges have special meanings (see section 2.6).
- Source Address
  This is the 12-bit device address of the sender of the message, in the range '000'H to 'FFF'H. The receiving device can read this from its message receive buffer after reception. Certain addresses and address ranges have special meanings (see section 2.6).
- Message Type and Length
  Two 4-bit fields normally used to indicate the type/length of the message. These bits are transported transparently by Conan.
- Data 0 to 15
  The message data. All 16 bytes are always transported. The Message Length normally indicates how many of the 16 bytes are actually valid for the message. The sending device writes this into its message transmit buffer for transmission. The receiving device can read this from its message receive buffer after reception.
- CRC
  A 16-bit Cyclic Redundancy Check value used to verify that the control frame has been transported without error. The CRC is generated by Conan automatically on message transmission and checked by Conan automatically on message reception.
- ACK/NAK
  Acknowledge and Not Acknowledge (2-bits each) indicate successful message transmission. The use of separate ACK and NAK flags allow reliable point-to-point *and* broadcast message transport. The flags are automatically filled by the destination device(s) (if present) and read by the sending device.
- Reserved : 10 bits reserved.

2.2 Source Data Ports and Formats

To maximise the application versatility of the chip, six source data ports are provided. This means that a product which needs to process source data for more than one internal source or destination can do so with only one Conan. These source data ports can transfer 8, 16, 24 or 32 bit source data, left or right adjusted, in to and out of the device.

The source data ports provide access to the source data in the network bit-stream. Data can be input through three serial inputs (SR0, 1 & 2) and output through three serial outputs (SX0, 1 & 2). All six ports use a common frame synchronisation FSY and a serial bit clock SCK. FSY and SCK may be set as either inputs or outputs, depending on the external hardware. If they are configured as inputs, then the data source(s) must be clocked by RMCK to be synchronised in frequency to the network bit-stream (although not necessarily in phase).

With the control bits in the Source Data Port Control register, a variety of source data formats can be selected. Some common formats are shown in the following diagram:



Some Common Source Data Formats

The register settings for these modes are given in section 4.1.4.

The way that source data presented to the input ports is routed to the outgoing bit-stream, and the way the incoming bit-stream is routed to the output ports is determined by the source data routing, is explained later.

2.3 Source Data Port Synchronisation

In a Conan network, the timing of all slave nodes is synchronised to one System Master. A slave node (e.g. a CD-Changer, Amplifier) must synchronise itself to the network by recovering the clock from the received bit-stream. The synchronisation of the source data between the I.Cs and Conan inside a product is achieved using the SCK, FSY and RMCK signals.

SCK is the 'bit clock' (also known as 'shift clock') for the source data. FSY is the 'frame synchronisation' which indicates 'left' or 'right' data. Conan's SCK and FSY pins can be configured as either inputs or outputs with the I/O bit in the Source Data Port Control register.

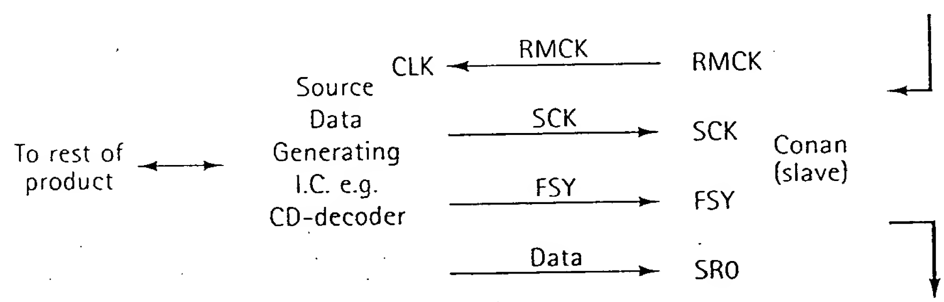
RMCK is the clock recovered from the network ('Received Master Clock'). The RMCK pin is an output only. The RMCK output frequency is always a (known) multiple of the sampling frequency (e.g. $384f_s$), set in the Clock Manager Control register.

2.3.1 Synchronisation in a Slave

A slave node can act as a source or destination (or both or neither) for source data. In a source node, the outgoing source data must be synchronised to the network. This means that the I.Cs in the source node which generate the source data (e.g. a CD decoder chip) must be locked to the incoming network clock in frequency (but not necessarily in phase). In a destination node, the I.Cs which process the source data (e.g. D/A converter) must be locked to the incoming network clock in the same way.

2.3.1.1 Source Example 1 (CD-Changer)

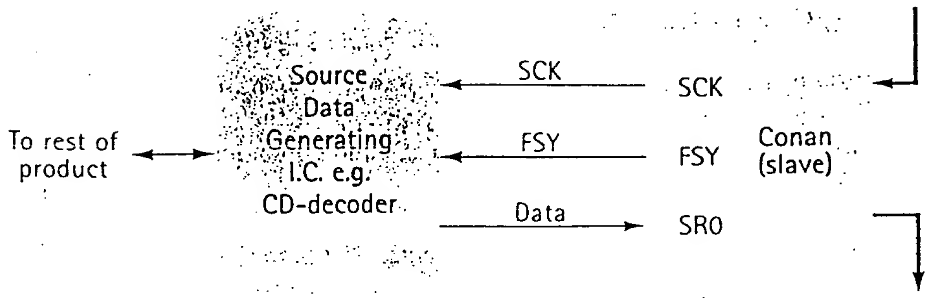
Here a CD-Decoder is clocked by RMCK from Conan. The CD-Decoder gives Data to Conan in synchronism with RMCK. SCK and FSY could be either outputs from the I.C (as shown) or inputs, depending on what the I.C. supports.



CD-Decoder clocked by RMCK

2.3.1.2 Source Example 2 (CD-Changer)

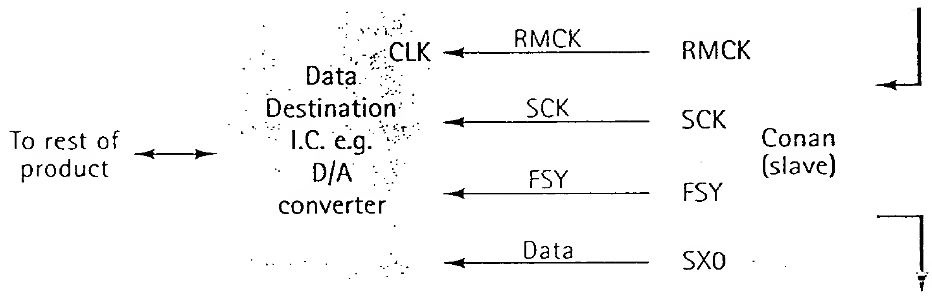
Here the CD-Decoder gives the Data to Conan in synchronism with SCK. RMCK is not used. In this configuration, SCK and FSY must be inputs to the I.C (as shown), so that the CD-Decoder can synchronise to SCK.



CD-Decoder clocked by SCK

2.3.1.3 Destination Example (Amplifier)

Here the D/A converter is clocked by RMCK from Conan. The D/A converter gets Data from Conan in synchronism with RMCK. SCK and FSY could be either inputs to the I.C (as shown) or outputs, depending on what the I.C. supports.



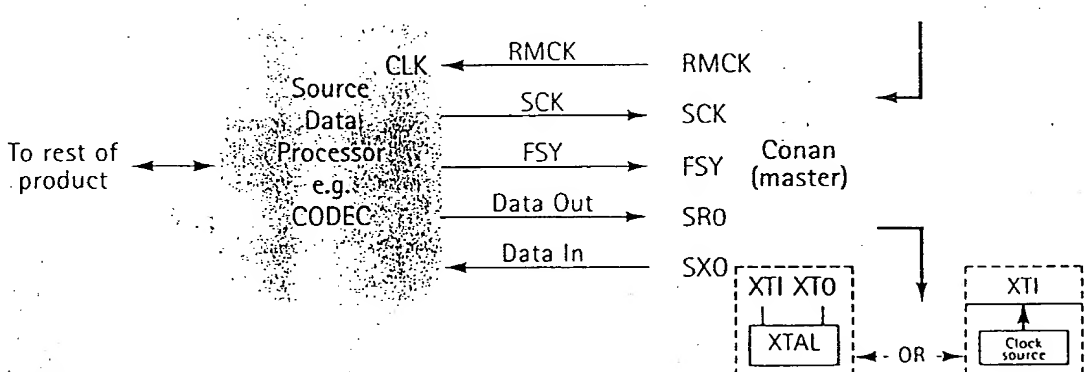
D/A Converter clocked by RMCK

2.3.2 Synchronisation in the System Master

The System Master generates the timing for the whole network. A System Master can act as a source or destination (or both or neither) for source data. For a source or destination node (or both) the I.Cs which process the source data must be clocked by RMCK derived either from a crystal or an external master clock.

2.3.2.1 Example (Radio/Head-Unit)

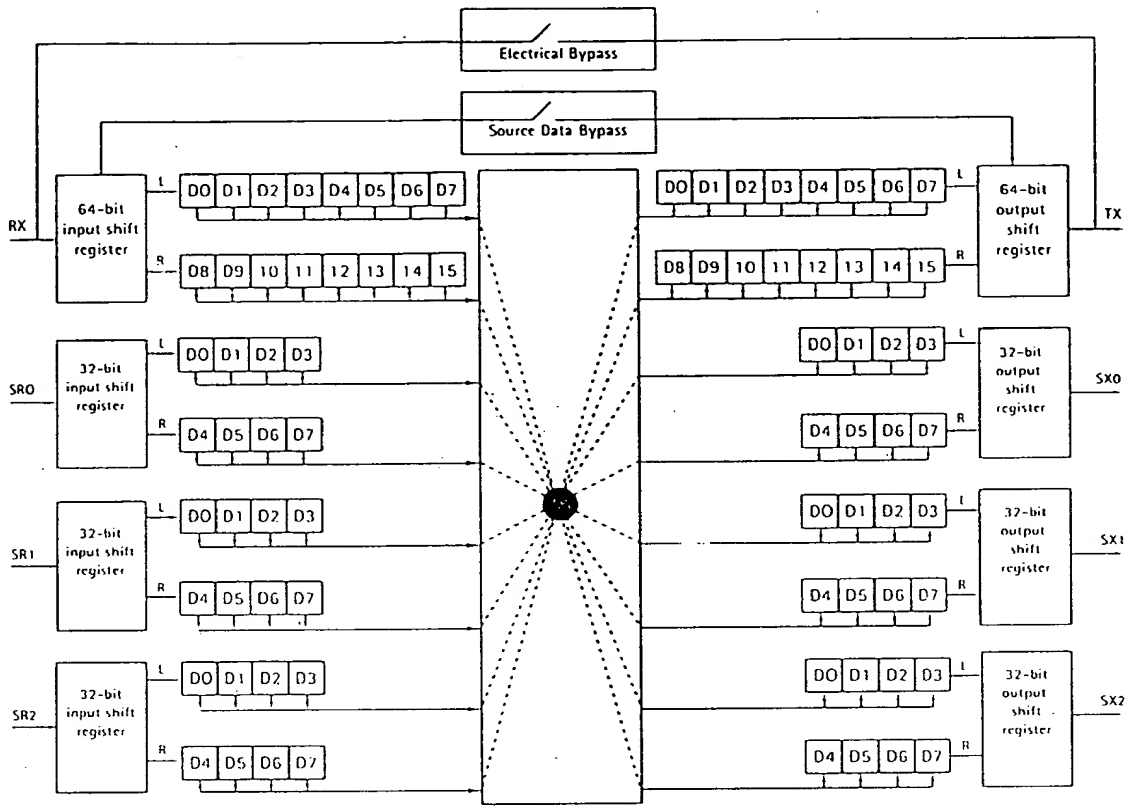
Here the CODEC is clocked by RMCK from Conan. The CODEC gives/gets Data to/from Conan in synchronism with RMCK. SCK and FSY could be either outputs from the I.C (as shown) or inputs, depending on what the I.C. supports.



Synchronisation in the System Master

2.4 Source Data Routing

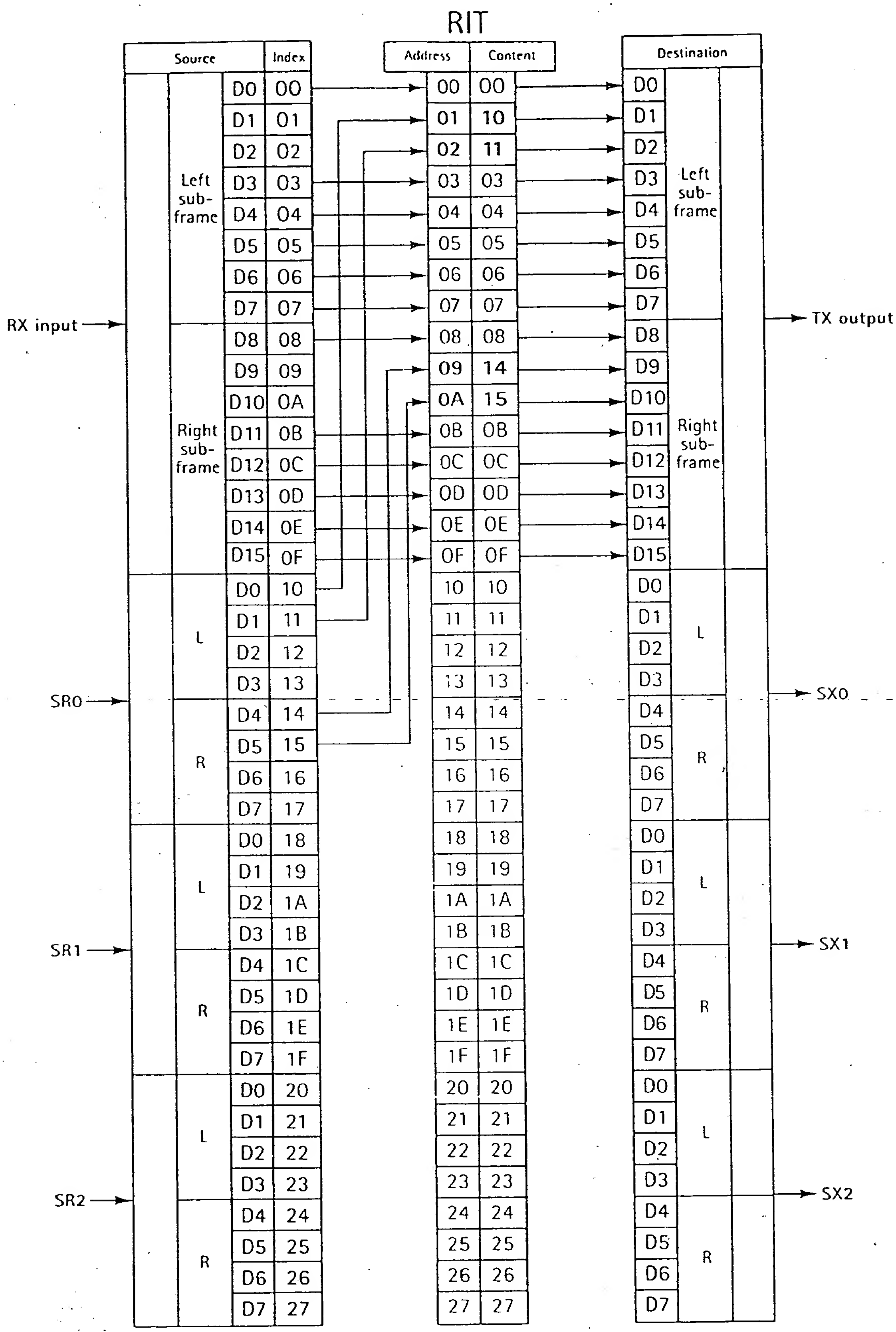
The Router provides an easy and flexible way of controlling the routing of source data. All source data bytes, coming either from the receiver or from the input ports can be re-arranged, mixed and re-directed to the transmitter or the source data output ports. All source data inputs are double-buffered to allow re-synchronisation. The following figure shows the source data inputs and outputs of the chip, with the router in-between.



Source Data Router

An incoming (serial) sub-frame is shifted into Conan bit-by-bit into a shift register, copied into a source data input buffer, moved by the router into a source data output buffer, and finally shifted out from Conan bit-by-bit into the outgoing sub-frame. These shifting and moving operations delay the source data in the bit-stream as it passes through the node by a time equal to 4 Conan sub-frame times (approx. 45µS at f_s=44.1kHz). Nodes which do not modify the source data in the bit-stream (i.e. which are not sources of source data, e.g. Amplifiers) can reduce this delay to only 1 bit by activating Conan's source data bypass (SBY bit in the Transceiver Control Register).

The smallest amount of source data that Conan can manipulate individually is eight bits - a 'source data byte' (Dn). There are a total of forty source data bytes as possible inputs to the Router (sixteen from the frame received from the network, and eight from each of the three source data input ports) numbered sequentially from 00H to 27H. The destinations for these input source data bytes are the forty output source data bytes (similarly numbered) of the transmit frame and the three source data output ports. The router is controlled by the Routing Information Table (RIT), which has forty entries. To route the input source data byte Dn with the index X to the output source data byte with RIT address offset Y, simply write the value X into the address offset Y of the RIT.



Example of Router Operation (64 clock cycles per frame)

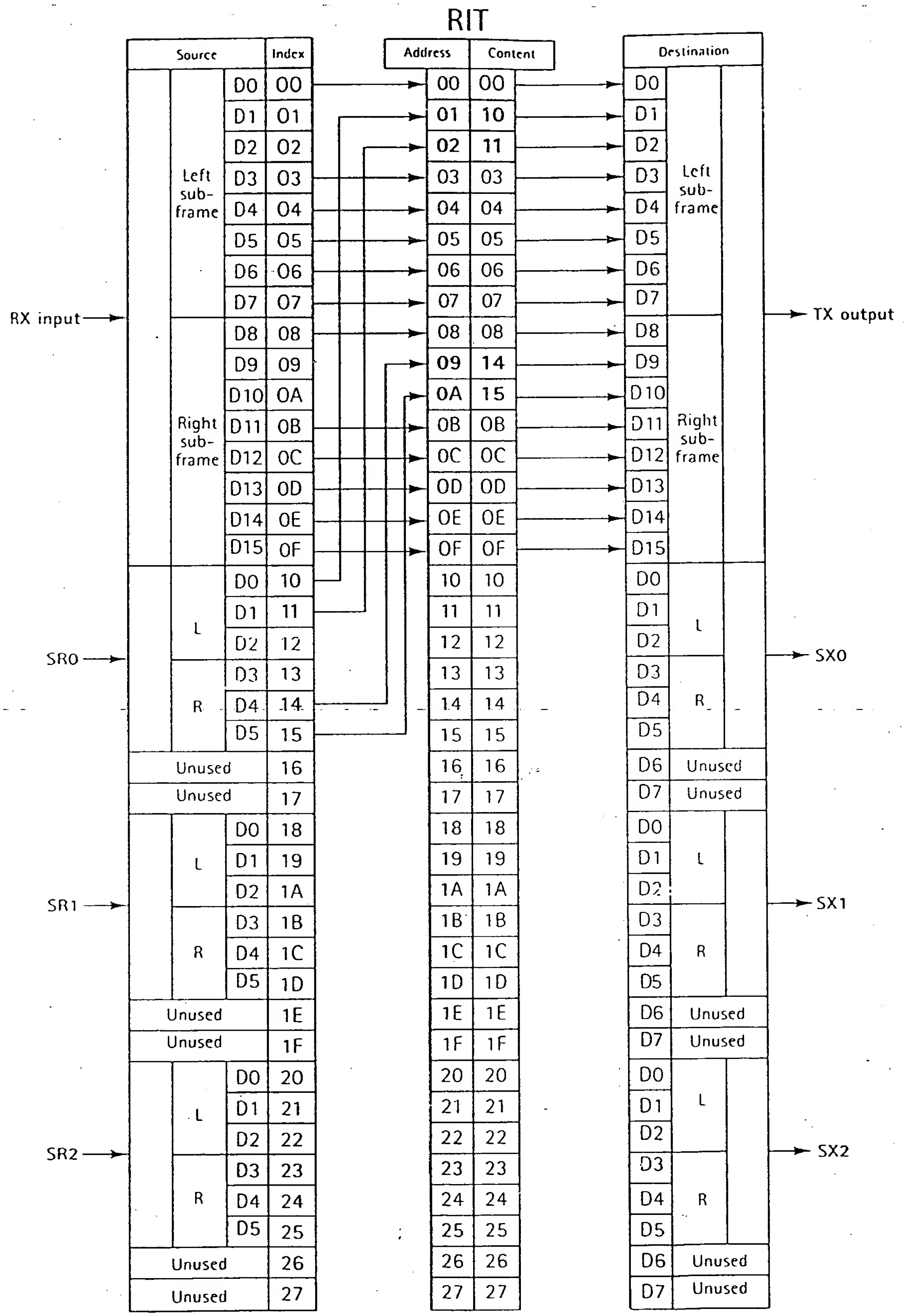
The example above shows a 16-bit stereo channel being routed from SR0 to the outgoing Conan bit-stream. In this case, the RIT offsets 1H, 2H, 9H & AH must contain 10H, 11H, 14H & 15H respectively. All other bytes of the incoming frame are routed directly out to the outgoing frame unchanged.

Source bytes with index 0H & 7H for the left sub-frame (and 8H & FH for the right sub-frame) contain the transparent channels, control and status bits, so would normally always be routed through unchanged.

After power-up or reset, the offsets 00H to 27H contain the values 00H to 27H respectively. All source data bytes received from the network are then routed directly out to the transmitter unchanged.

The description above for the source data routing applies where there are 64 clock cycles per frame. Conan can also be configured for 48 clock cycles per frame (with the Source Data Port Control Register), and then the following diagram applies.

50



Example of Router Operation (48 clock cycles per frame)

2.5 SPDIF Mode

Conan can support SPDIF (also known as AES/EBU or IEC-958) with its SPDIF mode. The features of this mode are:

- An SPDIF channel from a local SPDIF source may be presented to SR0.
- An SPDIF channel generated at one node may be transported transparently (without loss) on the network to all other nodes (up to 24 source data bits and the VUC bits).
- An SPDIF output is available from SX0 to present to a local SPDIF destination, even if there is no SPDIF channel being transported on the network (see 'SPDIF output from non-SPDIF data', section 2.5.3).

SPDIF mode is activated when the SPD bit in the Source Data Port Control register is set. Input SR0 and output SX0 will then receive and transmit source data in SPDIF format. The other source data ports (SR1, SR2, SX1, SX2) remain available to communicate non-SPDIF data, as configured in the Source Data Port Control register.

FSY and SCK must be synchronised to the SPDIF input, so they are automatically configured as outputs to maintain synchronisation requirements. Note that FSY and SCK are common for all source data ports. When Conan is in master mode, it can recover a clock from the SPDIF input on SR0, which it can then use internally (see Clock Management section 2.7).

The SPDIF receiver on SR0 checks for parity and bi-phase coding errors in the incoming SPDIF channel. If an error is detected, the validity bit ('V') is automatically set, so that an SPDIF destination could be muted. The SPDIF transmitter on SX0 generates the proper parity.

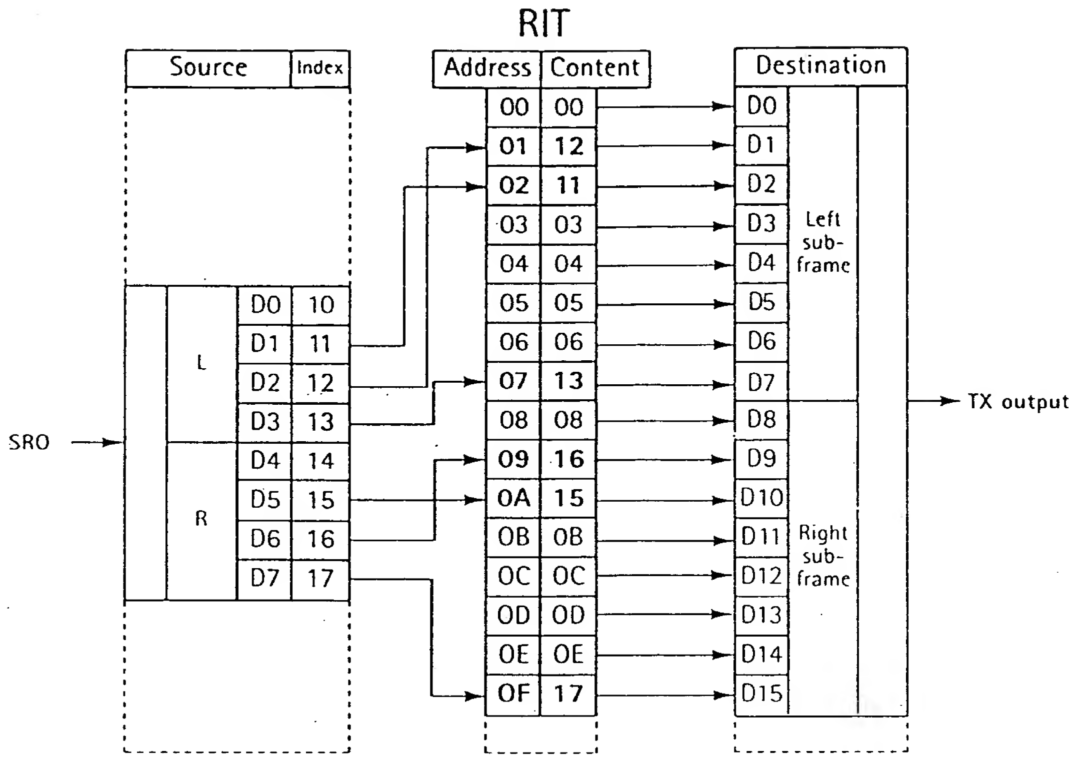
The channel status bits and transmitter block timing are controlled by Conan. The transmitter simply encodes data written to Source Data Port 0. When outputting an SPDIF channel recovered from the network, the channel status, user and validity information is contained in the data. When outputting an SPDIF channel recovered from non-SPDIF data, 'fake' channel status, user and validity bits may be read from the SPDIF Data Register.

In the SPDIF format, data is transported least-significant bit first. Conan's SPDIF ports take this into account and automatically change the order within the bytes to MSB first. However, the user has to ensure that the bytes are routed in the correct order through the RIT (see later). Both 16- and 24-bit SPDIF formats can be supported.

2.5.1 SPDIF Input and Transmission

If an SPDIF channel is input on SR0 for transmission on the network, the RIT must be programmed so that source data bytes of the SPDIF input are routed correctly into the outgoing Conan sub-frame, and that the VUC bits are inserted into byte 7 of the sub-frame.

For example, to transmit a 16-bit SPDIF channel on the network in the first two bytes of the Conan sub-frame, the RIT offsets 1H, 2H & 7H for the left sub-frame could contain 12H, 11H & 13H respectively. Likewise for the right sub-frame, the RIT offsets 9H, AH & FH could contain 16H, 15H & 17H (see example below).



Example of SPDIF Input and Transmission

2.5.2 Reception and SPDIF Output

If an SPDIF output is required from SX0, the RIT must be programmed so that source data bytes in the incoming Conan sub-frame are routed correctly to the SPDIF output, and that the VUC bits are taken from byte 7 of the sub-frame.

For example, to receive a 16-bit SPDIF channel which is occupying the first two bytes of the Conan sub-frame, the RIT offsets 11H, 12H & 13H for the left sub-frame should contain 2H, 1H & 7H respectively. Likewise for the right sub-frame, the RIT offsets 15H, 16H & 17H should contain AH, 9H & FH.

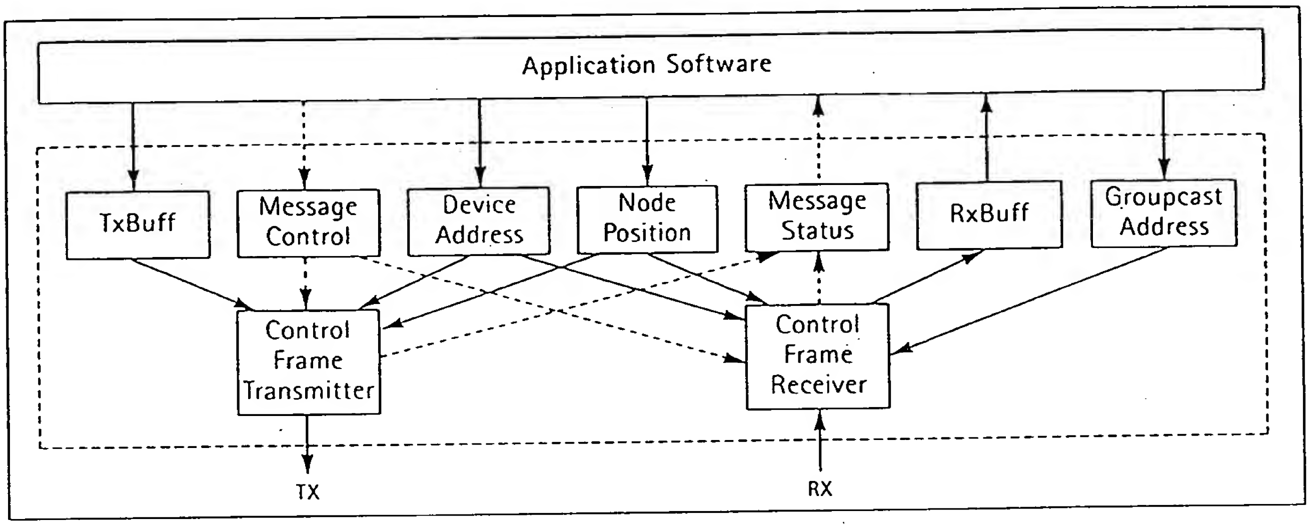
2.5.3 SPDIF Output from non-SPDIF data

If an SPDIF output is required from SX0, even though there is no SPDIF channel being carried on the network, 'fake' VUC bits can be inserted into the SPDIF output channel. The SPDIF Data Register (2CH) must be initialised with the desired VUC values (see section 4.1.3) and the contents of RIT offset 17H (which contains the preamble for the next left sub-frame) must be set to 28H, which then routes the fake VUC values into byte 3 of SX0. The contents of the SPDIF Data Register are processed by Conan and are made available in offset 28H for routing.

When this is done, each SPDIF sub-frame output from SX0 will have the VUC bits set as in the SPDIF Data Register. The bits in the register can be updated as fast as the control port will allow (up to approx. 300 times per second). Typically, this feature will be used to control the V bit to mute/demute an SPDIF destination.

2.6 Control Message Transceiver

The Control Message Transceiver is able to send and receive control messages to/from other devices on the network. It has a number of registers and message buffers associated with it - the Transmit Buffer (TxBuff), Receive Buffer (RxBuff), Message Control register, Message Status register, Device Address register, Groupcast Address register and Node Position register. These are shown below.



Control Message Transceiver Block Diagram

2.6.1 Address Initialisation

Following reset, Conan's device address is 'FFF'H, and it is able to send and receive messages. The device would normally then initialise its other address(es).

Each node actually has five addresses:- its 'device address', used for point-to-point messaging; its 'broadcast address' (fixed at '3C8'H) to receive broadcast messages; its 'groupcast address' to receive groupcast messages; its 'node position' to receive messages address by node position; and its 'set-position address' (fixed at 'F00'H) to receive the set-position command. These are all described in more detail later.

The device address and groupcast address must be initialised by the local application software. The node position address is initialised as part of the network start-up procedure.

2.6.1.1 Initialising the Device Address

There are two ways to initialise the device address:

- Using the automatic device address initialisation and verification mechanism built-into Conan
- Initialising and verifying the device address manually

Automatic Device Address Initialisation And Verification Mechanism

Write the new device address as the destination address field into TxBuff then set the SAI ('Start Address Initialisation') bit in the Message Control register (see section 4.1.6). Conan will check that the given address is unique on the network (by attempting to send a message to it), and if so, will write the verified address into its Device Address register (see section 4.1.12). The application may write a message into the data field of TxBuff before setting SAI, if needed.

Once the address is tested, the MTX ('Message Transmitted') bit will be set in Message Status register (see section 4.1.7) and an interrupt may occur depending on the Int Mode (see section 4.1.9). The TXR ('Transmission Result') bit reflects the result of the test and if set means that the address is unique in the system and is now the adopted address for this device.

If TXR is not set, the address is rejected and the device address remains set to 'FFF'H. Another device address may be tried by the application. Just as for any normal transmission, the RTI ('Reset Transmission Interrupt') bit must always be set ready for the next transmission.

Manual Device Address Initialisation And Verification

Simply write the new device address into Conan's Device Address register. This mechanism should only be used in fixed/closed systems where address conflicts cannot occur, or for software test/development purposes.

2.6.1.2 Initialising the Groupcast Address

To receive groupcasts, a device must write the least-significant byte of its groupcast address, value '00'H to 'FF'H (except 'C8'H) into the Groupcast Address register(see section 4.1.10).

2.6.1.3 Initialising the Node Position

This mechanism enables each device on the network to find out its position relative to the System Master. The System Master application software initiates the procedure by sending the following message: Destination Address = 'F00'H, Msg Type/Length = don't care, Data[0] = 03.

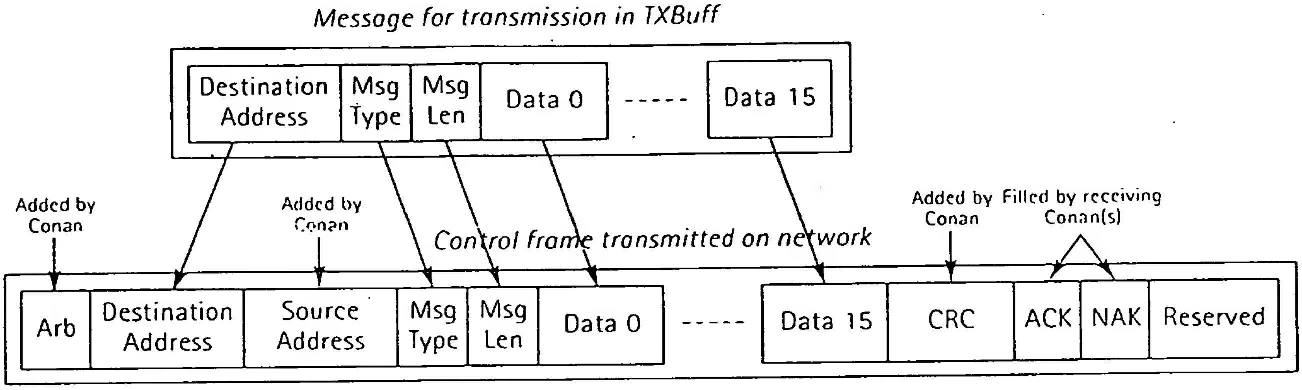
This message is passed around the network from device to device within a single control frame. Each device receives the message, stores its position in its Node Position Address register (see section 4.1.8) and passes the message on to the next device. This is all done automatically by Conan - no application software is needed in slave devices.

The procedure is complete when the message returns back to the System Master. At this moment, every device will know its position on the network, regardless of the transmission result.

2.6.2 Sending a Message

2.6.2.1 General Procedure

The general procedure to send a message is to write the message into the transmission buffer (TxBuff) and then set the STX ('Start Transmission') bit in the Message Control register. The control message transceiver reformats the message for transmission and sends the message with up to 5 retries, 10ms apart, if necessary. A gap of 10ms is also inserted between outgoing messages to prevent a node monopolising the control channel. The diagram below shows how Conan formats a message for transmission into the control frame.



Message Transmission

When the transmission is complete, MTX ('Message Transmitted') will be set and an interrupt may occur depending on Int Mode. TXR ('Transmission Result') is set if the transmission was successful and reset if the transmission failed. The condition must always be cleared by setting the RTI ('Reset Transmission Interrupt') bit in the Message Control register. A flow-chart for this sequence is given in section 3.2.4.

Conan can send a message in four different ways:

- point-to-point, to a device with a known address,
- broadcast, to all devices,
- groupcast, to a group of devices (e.g. all amplifiers), and
- node position addressing, to a device at a certain position on the ring

These different methods are described below.

2.6.2.2 Point-to-Point

To send a point-to-point message, write the device address of the destination into the destination address field of the Tx buffer, then initiate transmission as described in 'General Procedure' above.
The receiving Conan will set the ACK bit if it receives the message correctly, otherwise it will set the NAK bit. The sending Conan examines the ACK & NAK bits and generates a transmission result available in TXR.

2.6.2.3 Broadcasting

Broadcasting allows a message to be sent to all devices on the network. The broadcast address is '3C8'H. All nodes will receive messages broadcast to this address. This address is fixed within Conan so there is no software overhead involved in initialising the broadcast address. To send a broadcast message, write '3C8'H into the destination address field of the Tx buffer, then initiate transmission as described in 'General Procedure' above. The receiving Conan(s) will set the ACK bit if they receive the message correctly, otherwise they will set the NAK bit. The sending Conan may retry up to 5 times, 10mS apart. If a destination has already received the message (i.e. its Rx buffer already contains the message) then it will *not* set the NAK bit. The transmission result is available in TXR.

Broadcast messages have an ID byte inserted by Conan automatically at the end of their data field, which is incremented for each new broadcast message (not on each (re)transmission). The ID enables receiving Conans to determine automatically whether the message is new, or if it is a re-transmission of a message which they have successfully received already. The ID occupies data byte 15. This means only 15 data bytes (0..14) are available for message data.

If a Conan is triggered to send a message while a broadcast transmission from another Conan is in progress, it will wait until the broadcast is complete before attempting to send the message.

2.6.2.4 Groupcasting

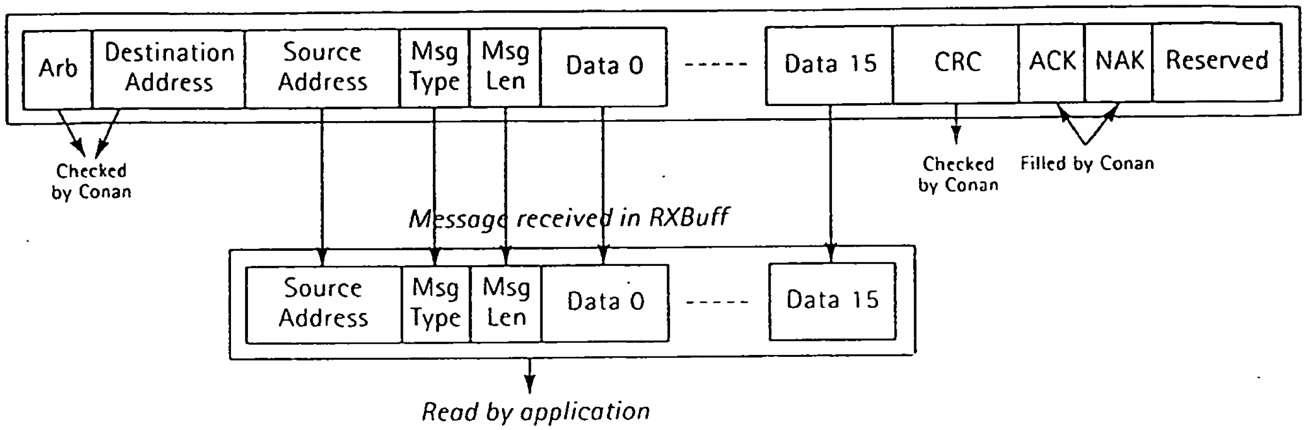
Groupcasting allows broadcasting to a particular group of devices which have the same groupcast address only. Groupcast Addresses are in the range '300'H to '3FF'H (excluding '3C8'H for normal broadcasts). To send a groupcast message, write the groupcast address into the destination address field of the Tx buffer, then initiate transmission as described in 'General Procedure' above. The transmission result is available in TXR. Groupcasting is the same as broadcasting in all other respects.

2.6.2.5 Node Position Addressing

To send a message to a node position, write the node position address (in the range '400'H to '4FF'H, where the least-significant byte is the node position of the destination device) into the destination address field of the Tx buffer, then initiate transmission as described in 'General Procedure' above. The transmission result is available in TXR.

2.6.3 Receiving a Message

The diagram below illustrates how a received control frame is formatted by Conan for passing to the application.



Message Reception

When a valid message is received, MRX is set in the Message Status register and an interrupt may occur depending on Int Mode. The condition should be cleared by writing a 1 to RRI ('Reset Receive Interrupt'). When the message has been read, the receiver buffer is re-enabled by setting RBE ('Receive Buffer Enable'). A flow-chart for this sequence is given in section 3.2.4.

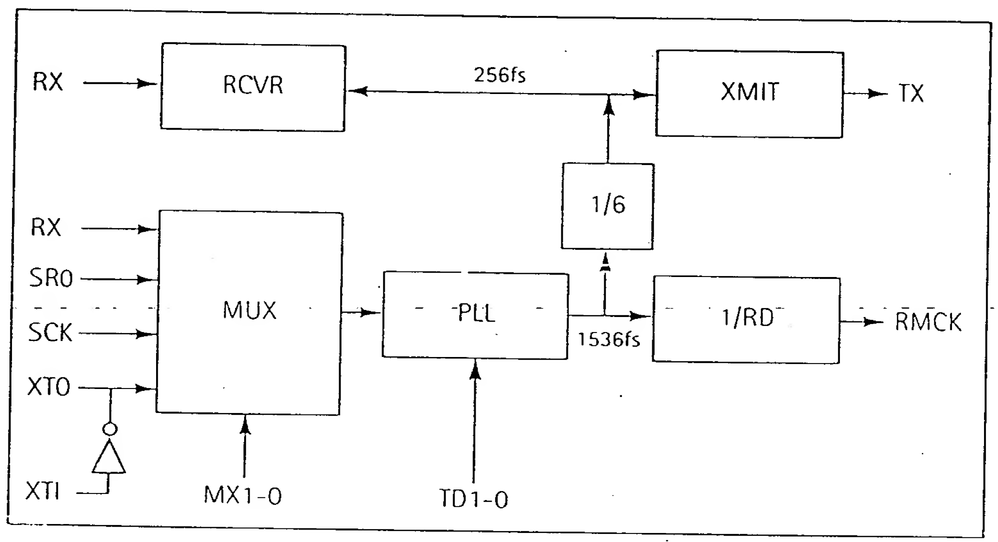
2.7 Clock Management

Conan can be configured by software as either a Master or a Slave. Conan needs a clock source to set the timing of its transmitter and to use internally.

In Slave mode, Conan's timing is derived from the incoming bit-stream. The network receiver recovers a clock from the incoming bit-stream using a low jitter, clock multiplying, phase locked loop (PLL). The PLL will lock to incoming bit-streams with sample rates of 30 to 50kHz. On loss of lock, the PLL is pulled to its lowest frequency (to minimise EMI).

In Master mode, Conan's timing may be derived from a crystal, SCK input or an SPDIF channel applied to SR0. The Oscillator circuitry allows a crystal to be connected directly to XTI/XTO, or some other clock source to be used. The PLL multiplies-up this timing source. The RX pin is over-sampled by a fast clock and data is recovered by a digital state machine.

The Clock Manager is controlled by the Clock Manager Control register and illustrated below.



Clock Manager Block Diagram

The PLL is enabled automatically on power-up, but can be disabled (pulled to its lowest frequency) to reduce EMI. If no transitions are occurring on the RX pin, the PLL is automatically pulled to its lowest frequency. A programmable divider in the PLL allows the frequency of a connected crystal to be $256f_s$, $384f_s$ or $512f_s$. The RMCK output is a programmable clock output, offering a divided down version of the internal $1536f_s$ clock generated by the PLL. A range of frequencies between $64f_s$ and $1536f_s$ are possible.

The PLL Mux selects one of four inputs for the PLL. As a slave, only the RX input may be used. As a master, there is a choice between the crystal oscillator (or other clock source applied to XTI), bit clock (from SCK input) and SPDIF data (from SR0 input). If the crystal is not selected, the oscillator will be disabled to reduce EMI.

The PLL lock state is made available to applications on the ERROR pin (pin 27), the ERR bit in the Transceiver Status Register and the POR/ERR bit in the Message Status Register. The time taken for the PLL to lock after receiving valid frames (or change in sampling frequency) is typically less than 10mS. Lock is declared if three consecutive valid left preambles are found. If two consecutive left preambles are not valid, loss of lock is declared.

2.8 Transparent Channels

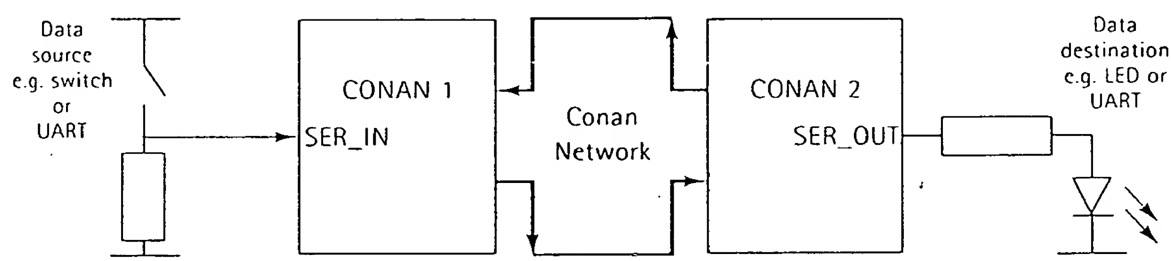
Conan can transport four channels of proprietary serial data on the network. The use of these channels is left open to System Integrators, who must define their own protocols for applications. Typical applications include the transport of raw control or status information, such as RS232-type data, VICS data, GSM data, etc.

The 4 transparent channels are transported independently in the 4 TC bits in the Conan sub-frame. Conan provides access to one of the four channels at a time (selected with the TC0-1 bits in the Transceiver Control Register) although any number of devices may monitor the same transparent channel. It is not possible for a device to send and receive on different transparent channels.

Data presented to the SER_IN pin of Conan in one device may be transported transparently to Conan in another device and output at its SER_OUT pin. The SER_IN pin is sampled twice every frame (i.e. 88.2 k-samples per sec at f_s=44.1kHz), on both FSY edges. Data is valid for reading from the SER_OUT pin on both FSY edges.

Asynchronous Transport

RS232-type data, up to about 19,200 baud, can be transported asynchronously with no hardware or software overhead. Conan 1 over-samples the signal on its SER_IN pin, and inserts the bit sample into the selected TC bit in the Conan sub-frame. Conan 2 receives the sub-frame, reads the selected TC bit, and outputs it at its SER_OUT pin. The destination (e.g. UART) should over-sample the SER_OUT to recover the data. In both cases, the over-sampling helps to preserve the mark/space ratio of the original data.



Asynchronous Transport

Of course, they may be more than just the one destination Conan shown above. Any number of Conans may monitor the same transparent channel.

Synchronous Transport

The data throughput may be increased up to a maximum of 88.2 kbps by using synchronous data, i.e. where the data is presented to the SER_IN pin in synchronism with the clock recovered from the network, and where the sample is taken when the bit to be sampled is known to be stable. In this case, additional hardware (a data synchroniser) will be needed.

The data synchroniser must hold the data given to it from the microprocessor in a latch, and present it to SER_IN on the FSY edge (for timing diagrams, see section 4.5.2.1).

2.9 Reset

Conan can be reset by hardware or software in 3 different ways:

- Power-on reset: when power is applied to Conan, it resets itself automatically.
- Hardware reset: the /RST input (pin 23) is an active-low, level-triggered TTL input. A microprocessor can reset Conan by pulling the pin low (<0.8V) for (at least) 2µS, then setting it high again.
- Software reset: by writing a 1 to the RES bit in the Message Control register

Selection of I2C or SPI

I2C or SPI is selected during a 'hard' reset (not a software reset). I2C mode is selected by holding SCL high during the reset. SPI mode is selected by holding SCL low during the reset. The selection is completed when the reset pin goes back high. The microprocessor must wait for the POR interrupt before starting to communicate on the selected I2C or SPI.

After Reset

Immediately following any of the above reset conditions, the ERR bit in the Transceiver Status Register is set (if the PLL has lost lock), the POR/ERR bit in the Message Status register is set to indicate power-on, an interrupt is raised which cannot be disabled (/INT goes low), and the Device Address is set to 'FFF'H. The crystal oscillator is disabled on reset (see section 4.1.5).

The POR interrupt must be cleared by setting the RPI bit in the Message Control register. This will clear the interrupt (/INT goes high) and clear the POR/ERR bit.

Revision Code

A 1-byte revision code is provided, so that application software can distinguish between different versions of the Conan.

It can be read by a microcontroller from Data 0 (register number 0xAD) in TxBuff. It is only available for reading immediately after resetting the chip and *before* the first message is sent.

For the Conan Engineering Samples, the revision byte was 0. For the Conan Production Devices, the revision byte is 1. Future versions will have different values.

2.10 Error Handling

Conan is able to detect errors and report them to the application, including:

- bi-phase coding or parity errors in the Conan sub-frame (from the network bit-stream)
- loss of lock to the incoming bit-stream from the network
- loss of lock to an incoming SPDIF channel on SRO

Errors are signalled with the ERROR pin (pin 27), the ERR bit in the Transceiver Status Register and the POR/ERR bit in the Message Status Register (depending on Int Mode). Which of the above errors are actually reported is selected with the ME, MDL and MSL bits in the Transceiver Status Register.

The ERR and POR/ERR bits are latched. If an error condition sets them, they stay set until they are cleared. The ERR bit is cleared by writing a zero to it. The POR/ERR bit is cleared by writing a 1 to the RPI bit in the Message Control register.

The ERROR pin is not latched. The pin goes high if an error condition occurs, and remains high for as long as the error condition exists. If a bi-phase coding or parity error occurs, the ERROR pin is high for the duration of the next sub-frame. If that sub-frame has no error, ERROR will go back low.

2.11 Interrupt Handling

The three events which can cause an interrupt are:

- Error (highest priority)
- Message transmission completed
- Message received

The /INT output from Conan is a TTL level, active low, open drain output. When an event occurs, the /INT pin goes low, and stays low until the interrupt is cleared. The events that trigger an interrupt can be selected with the Interrupt Mode register (see section 4.1.9). The modes are:

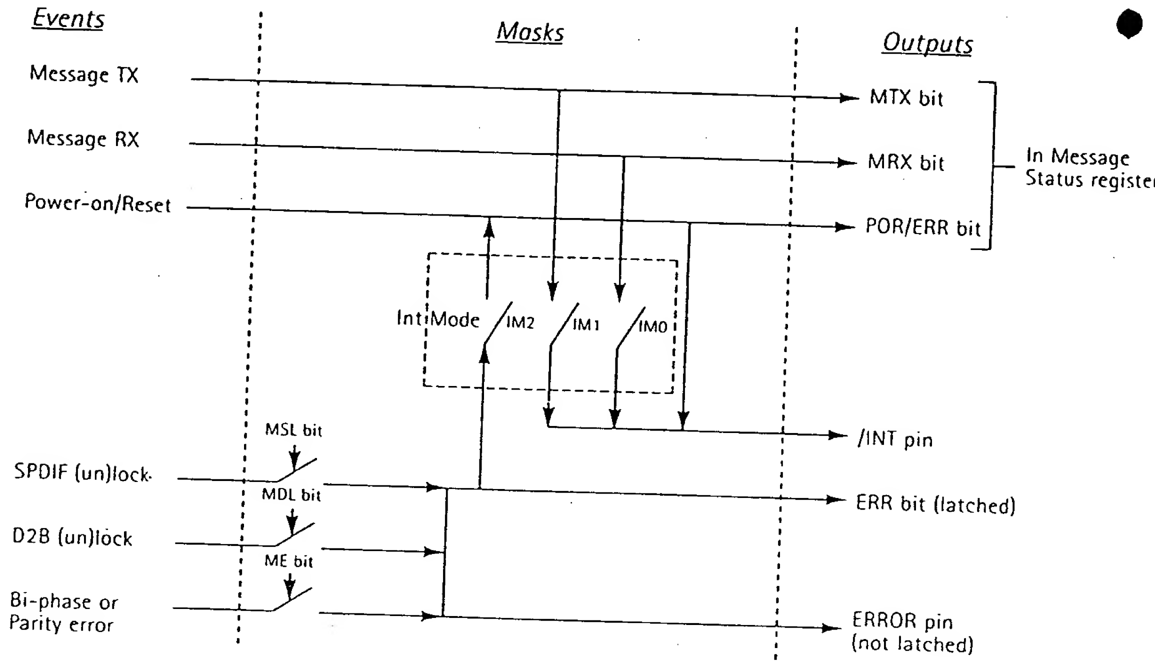
- Interrupts disabled (except power-on reset) - must poll the registers instead
- Rx and Tx interrupts only enabled
- Rx, Tx, and Error interrupts enabled
- Error interrupt only enabled

Where the error interrupt is enabled, the error types can be masked using the bits in Transceiver Status Register (see section 4.1.2). A flow-chart is given in section 3.2.4.

Rx and Tx interrupts are queued behind an Error interrupt. If an interrupt occurs due to an error and the Rx or Tx condition becomes true before the error interrupt has been cleared, then as soon as the Error interrupt is cleared another interrupt will follow to indicate the Rx or Tx condition.

If an error occurs whilst a Tx or Rx interrupt is active, then the error flags are set (ERR in the Transceiver Status Register and POR/ERR in the Message Status Register) to inform the application. Interrupts will recur until the ERR bit is cleared.

The diagram below shows which events, when enabled or disabled by bit masks, result in which outputs.



Events, Masks and Outputs

Where the error interrupt is enabled, when clearing an interrupt caused by Rx or Tx, the microprocessor should always also clear the error interrupt by setting the RPI bit, otherwise subsequent interrupts will be blocked.

For persistent errors such as loss of lock,

- disable interrupts (either at the microprocessor, or set the Interrupt Mode Register to 00)
- poll the ERR bit in the Transceiver Status Register until lock is re-established (look for a transition from 1 to 0). After each read, clear the bit.
- re-enable interrupts

CLAIMS

1. A local communication system comprising a ring network conveying source data in both variable rate and fixed rate channels, by means of a regular frame structure, each frame providing a fixed number of source data fields, wherein each fields can be reserved dynamically to form part of a fixed-rate channel using the same fields in each frame, and at other times can be allocated to form part of a variable rate channel for irregular data packets.

2. A system according to claim 1, wherein blocks for fixed rate data are allocated starting from one end of the frame, while fields for variable rate data are allocated starting at the other end of the frame.

3. A system according to claim 1 or 2, wherein successive frames are grouped into blocks, and each variable rate channel occupies the same fields through all frames of a block, fields being reallocated to provide variation of channel width only at the start of each block.

4. A system according to claim 1, 2 or 3, wherein a block header is transmitted to reserve a variable rate channel of a specified width for a plurality of

successive frames.

5. A system according to claim 4, wherein said block header occupies one or more fields of the channel for at least the first frame of the block.

6. A system according to claim 4 or 5, wherein at the start of a block, each channel's block header occupies one or more fields which can be located in the frame with reference to the widths of other channels.

7. A system according to any preceding claim, wherein each variable rate channel comprises a selection of fields fixed over a predetermined sized block of frames, the width of all such channels being specified in the source data fields of the first frame or frames of each block.

8. A local communication system comprising a ring network conveying source data in both variable rate and fixed rate channels, by means of a regular frame structure in which certain portions of each frame are reserved for said fixed rate channels, whether or not said fixed rate channels are in use, and certain other portions of each frame are available for said variable rate channels, and a control mechanism is provided for allocating said variable rate portion dynamically between

different channels.

9. A system according to claim 1, 2 or 8, wherein the frame rate is synchronised with one or more digital audio data sources, for which source data is carried in the fixed rate portions of each frame.

10. A system according to claim 1, 2, 8 or 9, wherein each frame conveys control bits forming part of a control message frame transmitted over plural frames.

11. A local communication system comprising a synchronous ring network conveying source data in a fixed rate channel over one segment of the ring and while said fixed rate channel is multiplexed with variable rate channels over another segment of the ring.

12. A system according to claim 11, wherein said multiplexed fixed rate channels and variable rate channels comprise different respective portions within a regular frame structure on said other segment of the ring.

13. A fibre optic local communication system, for example according to any of claims 1 to 12, suitable for in-vehicle entertainment, communication and/or navigation purposes, having an overall source data capacity greater

than 10 Mbps, the fibre optic channel conveying 4B5B or 8B10B encoded data.

14. A system according to any preceding claim, wherein variable data source data channels are mapped on to the network in asynchronous transfer mode packets.

15. A fibre optic local communication system, for example according to any of claims 1 to 12, suitable for in-vehicle entertainment, communication and/or navigation purposes, having an overall source data capacity greater than 10 Mbps, the source data comprising variable data rate audio and video data, carried by asynchronous transfer mode (ATM) packets.

16. A system according to claim 15, wherein the headers and data fields of ATM packets do not necessarily consist of 5 bytes and 48 bytes respectively.

17. A local communication system substantially as described herein.